MULTI-PHYSICS HIGH RESOLUTION TOPOLOGY OPTIMIZATION FOR AEROSPACE STRUCTURES

A Thesis
Presented to
The Academic Faculty

By

Ting Wei Chin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

May 2019

MULTI-PHYSICS HIGH RESOLUTION TOPOLOGY OPTIMIZATION FOR AEROSPACE STRUCTURES

Approved by:

Dr. Graeme J. Kennedy, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Brian J. German
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Julian J. Rimoli
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Claudio V. Di Leo
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kai A. James
Department of Aerospace Engineering
University of Illinois at Urbana-Champaign

Date Approved: March 19, 2019

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my advisor Dr. Graeme J. Kennedy for taking a chance on me. His guidance and support meant that I was able to fully immersed myself in research. Also, his broad scope of interest allowed me to pursue and investigate different computational areas that led to the work in this thesis. I would also like to thank the members of my thesis committee, Dr. Brian German, Dr. Julian Rimoli, Dr. Claudio Di Leo and Dr. Kai James for taking time out from their schedule to serve on the committee. Their suggestions and comments are invaluable to this work. I would also like to acknowledge Justin Gray, who provided guidance on problem design for this work.

I would like to thank all the friends I have met at the Aerospace Engineering Department. Special thanks to Komahan, whose fervent interest in everything computational inspires me; Jan, who turns out to have similar interests as I do both in and outside of research; Mark, who is always so helpful and asks for nothing in return; Chun Rong, a fellow Singaporean who works so hard for his family back home; Nicholson, who is great to talk and bounce ideas off each other; Last but not least, Kevin, who goes the extra mile to help anyone who asked. I would also like to thank all the people I have met along the way both in and outside of campus for making my journey an enjoyable one.

Lastly, I would like to thank my parents, Kai Sun and Sophie, for their constant unwavering support and emphasis for me to pursue my passion. Thanks to my sister, Lyanne, who is also nearing the end of her PhD journey, for her constant encouragement and making the effort to get together during the holidays.

This research was supported by NASA through the Transformative Tools and Technologies program with grant number NNX15AU22A under the direction of technical monitor Steve Massey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Advancements in multimaterial additive manufacturing have the potential to enable the creation of topology optimized structures with both shape and material tailoring. These are extremely useful in creating designs for multi-physics applications where engineering experience may be lacking. These include designing aerospace structures that are subjected to elevated temperature environment, where mechanical and thermal loads are present or designing structures for strength and avoiding low natural frequency resonance. Multi-physics analysis and multimaterial design parametrization present additional complexity and technical challenges to overcome for large-scale designs. Design and analysis using large-scale uniform meshes is computationally expensive due to the large number of degrees of freedom (DOFs). The same mesh resolution can be created through adaptive mesh refinement such that it has fewer DOFs. However, due to the complexity in creating these adaptive meshes, especially for higher order 3D designs, they are mostly confined to 2D topology optimization. Large-scale multimaterial design through Discrete Material Optimization (DMO) also results in numerous partition of unity constraints and a multimaterial design space that has more local minima than an equivalent single material design space. This work presents new techniques for obtaining large-scale 3D multimaterial, multi-physics designs. Adaptive mesh refinement and higher order design parametrization are introduced to obtain smooth features. The multi-physics capabilities of the method are demonstration in the form of thermoelastic topology optimization. Multimaterial designs using adaptive mesh refinement as well as higher order design parametrization with steady-state thermoelastic topology optimization are presented. Novel technique to accelerate large-scale natural frequency-constrained topology optimization design is also presented.

CHAPTER 1
BACKGROUND

Motivation

Topology optimization, first developed by Bendsøe and Kikuchi [1], is a useful tool for the design of aerospace structures [2, 3, 4], automotive structures [5, 6], extremal material design [7], meta-materials and meta-structures [8, 9], MEMS [10, 11] and fluid channel flow [12, 13]. The popularity of topology optimization is due to its ability to generate novel, optimized structures without geometric constraints. Topology optimization techniques require the domain geometry, prescribed loads and boundary conditions, and the design formulation consisting of the objective and constraints, and produce a design that is unconventional while satisfying the physics of the problem. Topology optimization techniques are especially beneficial when dealing with early phase design where design intuition might be lacking.

Recently, advancements in additive manufacturing technologies have further enabled engineering applications of topology-optimized designs [14, 15, 16]. Additive manufacturing for both polymer and metallic multimaterial structures has also been rapidly maturing, enabling simultaneous shape and material tailoring [17]. As a result, there is increasing interest in multimaterial topology optimization methods. Multiphase and multimaterial topology optimization techniques were first developed by Sigmund and Torquato [7] for thermal coefficient design and later extended by Gibiansky and Sigmund [18] to the design of extremal composites. Stegmann and Lund [19] developed an extension and generalization of these early multiphase methods, called Discrete Material Optimization (DMO), that has proven to be effective in numerous applications including multimaterial design and laminate design for composite

structures [20, 21, 22, 23, 24, 25, 26]. However, most DMO applications have focused on moderate-scale 2D problems, with fewer large-scale 3D applications [27]. This is due to the additional technical challenges beyond those encountered in conventional single-material topology optimization: (1) the design space is larger due to the addition of material selection variables, (2) a large numbers of sparse partition of unity constraints are often required for the most effective DMO parametrizations [21], making common topology optimization algorithms including the Optimality Criteria (OC) method and Method of Moving Asymptotes (MMA) incompatible with these problems, and (3) it is widely reported that the multimaterial design space is more multimodal than the equivalent single material design space [28, 29].

One class of multi-physics problem that will be a focus of this thesis is thermoelastic topology optimization [30]. Thermoelastic analysis is necessary when significant thermal loads are present from either internal or external sources. External thermal loads occur when aerospace structures are subjected to elevated temperature environments. For instance, primary and secondary structures are subjected to aerodynamic heating for supersonic flights and aerospace vehicles entering and leaving the atmosphere experience extreme temperatures [31]. Internal thermal sources can arise from the presence of heat-generating subsystems or devices, such as batteries, multifunctional structures with embedded energy storage [32, 33], or power sub-systems [34, 35, 36]. Thermal conditions generated from these external and internal sources are often design critical and must be considered early in the design process. The design and modeling of structures that operate in high-temperature environments requires analysis tools that can capture the physics of the thermoelastic problem, as well as mesh generation tools to create a suitable mesh given computer-aided design (CAD) geometry.

Furthermore, many aerospace structures are subject to high-vibration environments. The capability to evaluate the natural frequency of the topology optimized

designs is required when the final design experiences significant vibration [37]. To mitigate any unwanted effects, the minimum fundamental frequency of the design can be bounded from below. Its incorporation into large-scale 3D applications is predicated on two issues: (1) the ability to account for eigenmode switching as the designs change and (2) efficient method to obtain solve to the eigenproblem at each design iteration.

To address these issues for the design of high-resolution 3D domains, new design and analysis models as well as computational algorithms are needed. The thrust of this thesis is to develop new methods for multimaterial thermoelastic topology optimization that utilize adaptive mesh refinement to solve large-scale problems efficiently. While adaptive methods have been applied to thermoelastic problems [38, 39, 40], its applications to design optimization is limited. To design structures with unique thermal properties, the methods are implemented within a framework that includes a multimaterial formulation. Within this framework, a hierarchy of analysis and design meshes are created from an input CAD geometry by local refinement of an initial coarse quadrilateral/hexahdral mesh. To enable local mesh refinement, conforming quadtree/octree meshes are constructed on each element in the coarse quadrilateral/hexahedral mesh, thereby facilitating the construction of meshes for geometric multigrid and local mesh size control around design features of interest. An Sℓ_1QP method with a trust region globalization is implemented in parallel and is tailored to handle the large number of partition of unity constraints that arise from DMO parametrizations. Higher order analysis and design parametrization methods are also incorporated to allow for smoother designs and possible improvement to adaptive mesh refinement heuristics. To efficiently solve large-scale eigenproblems, the Jacobi–Davidson method is used [41, 42, 43] instead of the Lanczos method [44]. Jacobi–Davidson methods do not need highly accurate solution to the linear system, unlike Lanczos methods, thus making them potentially more computationally

efficient when paired with iterative procedures such as geometric multigrid methods that are already employed in other parts of the framework. Several eigenvector recycling strategies are formulated and evaluated as part of the effort to accelerate the Jacobi–Davidson method during the design process. Lastly, the evaluated natural frequency can be aggregated to form a single natural frequency constraint that is smooth and differentiable.

Large-scale Meshes and Adaptive Mesh Refinement

The creation of detailed optimized 3D designs requires finite-element meshes with tens of millions of elements or more. Several authors have developed methods to tackle large-scale 3D topology optimization problems for single material design [45, 46, 47], with the largest design case having 1.1 billion elements [47]. Aage, Andreassen, Lazarov, and Sigmund observed that topology optimization on large-scale meshes can capture detailed features which might otherwise be missing in lower resolution results. This scale has yet to be achieved for multimaterial problems, where the largest cases to date have on the order of one million elements [27].

Most meshes in topology optimization are generated with a uniform feature size since, in general, there is no *a priori* knowledge about the optimal topology. However, uniform refinement on large-scale meshes produces an inverse cubic dependence between element resolution and number of elements. Alternatively, adaptive refinement techniques can be used to refine the mesh to achieve sufficient resolution thereby breaking the inverse cubic relationship between resolution and number of elements. Figure 1.1 shows the number of elements in the mesh as a function of relative resolution for uniformly refined meshes (triangles) and for adaptively refined meshes (squares). These data points are taken from the work of [48, 49, 50, 46, 27] as well as five design cases that are presented here which are a combination of uniformly refined and adaptively refined meshes. Here, relative resolution is defined as cube root

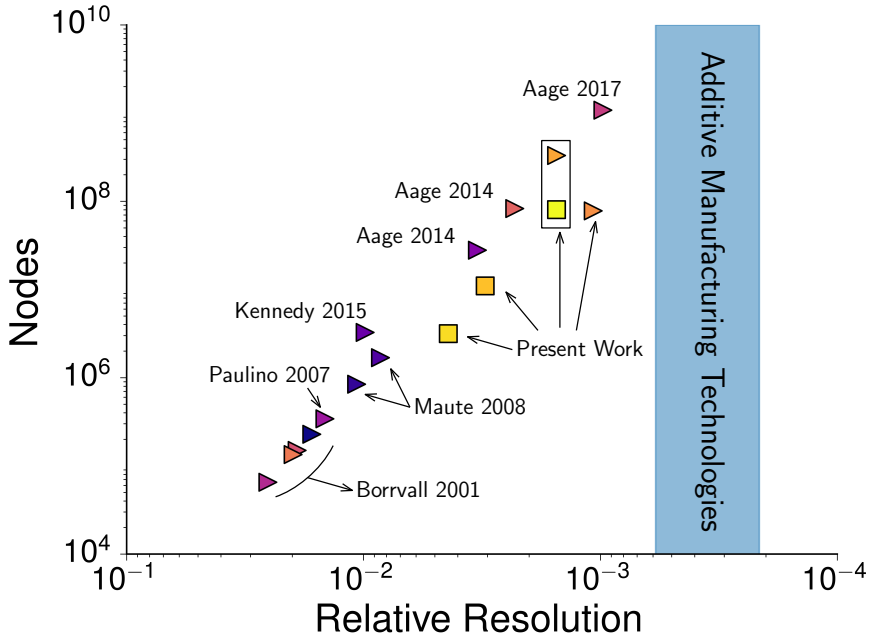


Figure 1.1: Number of nodes as a function of relative resolution for topology optimization results.

of the ratio of the smallest element characteristic volume to the characteristic volume of the domain. The blue colored band denotes the maximum relative resolution capability of current metallic additive manufacturing technologies [1]. Even the highest relative resolution mesh used for topology optimization has not yet achieved the relative resolution currently available through additive manufacturing. Comparing the two data points from the present work that are grouped together, both have the same relative resolution but the adaptive mesh refinement design point has far fewer nodes compared to the uniformly refined design, thereby requiring less computational resources.

Within the density-based topology optimization literature there are two common approaches to topology optimization with adaptive mesh refinement: (1) link the design and analysis mesh together and perform the refinement steps simultaneously on

[1]Relative resolution data obtained from 3D Systems Direct Metal Printers Series using build volume and minimum feature size. For example, ProX DMP 200 has minimum feature volume of $2\times 10^{-13}\,\mathrm{m}^3$ and build volume of $1.96\times 10^{-3}\,\mathrm{m}^3$, resulting in a relative resolution of 4.67×10^{-4}

both, or (2) maintain a separate design parametrization mesh and refine only the analysis mesh. Bennett and Botkin [51] pioneered the first approach, using adaptive mesh refinement for shape optimization to control the accuracy of the finite element solution for planar structures using 2D triangular elements. Around the same period, Noboru, Chung, Toshikazu, and Taylor [52] used r- and h- refinement techniques to generate meshes adaptively with a mixture of 2D triangular and quadrilateral elements. Adaptive mesh refinement can also be used for shape optimization with multiple loading conditions as demonstrated by Botkin [53]. In this case, the refinement criterion is computed for each load case and the load case that causes the mesh to be most refined is chosen as the subsequent starting mesh. More recently, Costa Jr and Alves [54] proposed an algorithm that optimized the structural layout using a h-adaptive approach with improved definition of material boundary. Instead of applying adaptive mesh refinement based on an element-wise error norm, Stainko [55] used a filter to identify the interface between material and void and refined elements along this interface since elements within a void/solid region is rarely influenced by optimization on a finer grid.

Several authors have tried separating the design and analysis mesh. Maute and Ramm [56] demonstrated that adaptively refining 2D quadrilateral element required less numerical effort but produced similar resolution compared to an uniformly refined mesh for topology optimization of surface mesh. Maute and Ramm [56] also introduced the concept of adaptively changing not only the size of the element but also the orientation of the finite element mesh which reduces the number of optimization variables. Similarly, these techniques are applied on a combined shape and topology optimization of shell structures [57]. Other authors have characterized the design mesh as a density point grid instead of having a mesh similar to one used in finite element analysis. Guest and Smith Genut [58] used adaptive mesh refinement on a design variable field consisting of nodal points that correspond to locations on the finite

element mesh. Using a Heaviside projection method, the authors were able to reduce the computational time as well as control the minimum feature size of the structural layout. Similarly, Wang, Kang, and He [59] carried out adaptive mesh refinement on the density point grid with a fixed finite element mesh throughout the process. Using a filter, the cut-off radius is adaptively modified to achieve structures with smooth boundary interface. More recently, Lambe and Czekanski [60] employed a continuous material density field and used adaptive mesh refinement to identify and refine along the boundary of the structural layout without any numerical artifacts such as checkerboarding.

Different heuristics can be used to drive the adaptive mesh refinement process. These heuristics often involve estimating the error in each element. One of the more popular error estimation technique was introduced by Zienkiewicz and Zhu [61], which uses the norm of the difference between interpolated stresses and the stresses computed from the finite element solution. The popularity of this approach stems from its ease of implementation in existing finite element solvers. Another common refinement heuristic is feature-based techniques such as the case of Stainko [55] who use a filter to determine elements along the solid-void boundary and refine them. Bruggi and Verani [62] combined two different refinement heuristics for topology optimization: one based on compliance and a second based on element density. For a compliance objective, they showed that using a combination of these two indicators produced a structural layout with fine interface between the solid and void region and a better design objective compared to an adaptive strategy that used only a density-based indicator. de Sturler, Paulino, and Wang [63] also mentioned the need for a dynamic adaptive mesh refinement strategy that allows for both refinement and coarsening of the mesh. They found that only then will the optimizer generate structural layouts that are equivalent to the optimal design on an uniformly fine mesh. Wang, Kang, and He [64] used an strain energy error indicator to refine its displacement field while

a functional gray transitional region measure (GTR) to refine the density field.

While some works have addressed large-scale topology optimization for single material [46, 47], large-scale designs involving multimaterial topology optimization have not been addressed, with the largest having $\mathcal{O}(10^6)$ elements [27], around 3 orders of magnitude smaller than the design presented in [47]. Also, the application of adaptive mesh refinement with multimaterial design is limited. Thus, part of this thesis is to identify and present methods that enables large-scale multimaterial topology optimization with adaptive mesh refinement.

Multimaterial Thermoelastic Topology Optimization

Structural topology optimization creates a structural layout for the given mechanical loads and boundary conditions. As the design evolves, the loading on the domain does not change. On the other hand, due to the multi-physics nature of thermoelastic topology optimization, there are not only mechanical loads present, there are also design-dependent loads, due to thermal expansion, that appear due to the presence of structure. As the optimizer changes the design, the combined mechanical and thermal loads on the domain also evolves, making thermoelastic topology optimization more difficult.

Thermoelastic topology optimization was first studied by Rodrigues and Fernandes [65] who used a homogenization approach to formulate the thermal stresses. Li, Steven, and Xie [66] used evolutionary structural optimization (ESO) technique to minimize the displacement of the structure. For weakly-coupled thermoelastic problems, Cho and Choi [67] formulated a coupled field adjoint sensitivity analysis method. Using the level set method, Xia and Wang [68] minimized the compliance with a mass constraint using thermoelastic formulation. Sigmund and Torquato [7] examined the design of composite microstructures using materials of vastly different thermal properties. Gao, Xu, and Zhang [25] used multimaterial parametrization to minimize the

compliance of both 2D and 3D structures for a combination of steady temperature fields and mechanical loads. Deaton and Grandhi [69] applied thermal loading for design of shells that do not accommodate thermal expansion, thus restricting the design space and creating different designs. Deng and Suresh [70] applied thermoelastic topology optimization with linear buckling and compliance constraints on different domains to illustrate the changes to the design due to the thermal loading. Also, Stanford and Beran [71] developed a framework to account for design-dependent loads such as aerodynamic, inertial, elastic and thermal loads to optimize a metallic panel subjected to different flutter and buckling considerations. In addition, Jog [72] extended thermoelastic topology optimization to include nonlinear thermoelasticity effects. Recently, many authors have pointed out the deficiency in using compliance as an objective for thermoelastic topology optimization. Through the works of Pedersen and Pedersen [73, 74], it was identified that minimum compliance and maximum strength objective do not result in equivalent optimal designs for the case of thermoelastic topology optimization. Deaton and Grandhi [75] also describes the need to consider topology optimized designs based on thermal stresses since it is a primary cause of failure in thermal structures.

Multiphase topology optimization was first developed by Sigmund and Torquato [7] for thermal coefficient design and later by Gibiansky and Sigmund [18] for extremal composite design. Discrete Material Optimization (DMO) was later developed by Stegmann and Lund [19] as an generalization of early multiphase methods and has subsequently proven to be effective in numerous applications [24, 25, 26]. Hvejsel, Lund, and Stolpe [22] applied the DMO parametrization to laminate composite design and added a quadratic penalty constraint to penalize intermediate designs. Other authors have developed schemes that are extensions to the DMO approach. Hvejsel and Lund [21] used SIMP or RAMP penalization modified for multiphase design problems as design-dependent weights. They also formulated sparse linear constraints such that

only a candidate material or void is chosen in the subdomain. Kennedy and Martins [24] developed a method for discrete ply angle selection of laminated composites by relaxing the constitutive matrices expressions to one that used continuous ply angle variables. An exact l_1 penalty function is used in order to obtain a discrete design. Other authors [76, 77] extended DMO to include thickness variables in the optimization of laminated composites by assigning an additional design variable known as the topology variable (DMTO). The topology variable takes the value of 1 if a material is selected and 0 if there is a void. Others have developed alternative methodology to the multimaterial design problem. Bruyneel, Duysinx, Fleury, and Gao [78] and Bruyneel [79] demonstrated the use of shape functions with penalization (SFP) to perform ply selection in laminated composite design problems. The formulation for SFP can be extended to different number of plies in the composite design by selecting appropriate shape functions and the number of design variables for each ply selection is fewer than that of DMO. For level-set method, Wang, Luo, Kang, and Zhang [80] proposed using multimaterial level set framework (MMLS), where M candidate materials are represented by M level set functions. James [81] recently applied the SFP parametrization to a multimaterial problem with a constraint on the cardinality set of candidate materials. Lastly, Kennedy and Chin [82] presented a sequential convex optimization method accelerates the solution to minimum compliance design of multimaterial problems through the use of Hessian-vector products to solve the convex subproblems.

There are few works that combined both multimaterial and thermoelastic topology optimization. Sigmund and Torquato [7] first examined the design of composite microstructures using materials of vastly different thermal properties to create a structure with novel thermal expansion properties. Sigmund [83] extended their approach to the design of multi-physics actuators. Gao, Xu, and Zhang [25] used multimaterial parametrization to minimize the compliance of both 2D and 3D structures for a

combination of steady temperature fields and mechanical loads. More recently, Chin and Kennedy [84] presented results on stress-constrained, mass minimization results for 2D structures. More recently, Kang and James [85] used multimaterial topology optimization to create 2D designs minimized for both structural and thermal compliance with constraints on the material specific maximum temperature limits. Based on literature, 3D multimaterial design with both structural and thermal constraints have yet been addressed. Thus, part of the the thrust of this thesis is to create methods to tackle this type of problems.

Higher Order Analysis and Design Parametrization for Topology Optimization

Higher-order finite element methods have been demonstrated to be superior to low-order finite element methods for linear analysis in several ways, namely faster convergence in terms of the energy norm, more accurate results for the same number of elements [86]. The application of higher-order finite element in topology optimization was first introduced by Díaz and Sigmund [87] and Jog and Haber [88] to avoid numerical artifacts such as checkerboarding when solving the structural analysis problem. Depending on the density penalization, biquadratic elements might not completely eliminate checkerboarding despite improved stability [87] and alternatives such as the use of use of density filters [11, 89] or sensitivity filters [90] in conjunction with linear finite elements has been introduced to increase numerical stability. Matsui and Terada [91] proposed a method called continuous approximation of material density (CAMD) that used higher order displacement field in conjunction with bilinear shape functions for both 3/6-noded triangular elements and 4/8-noded quadrilateral elements. Kang and Wang [92] constructed the density field through a non-local higher order Shepard interpolation scheme. The design variable points are freely distributed within the domain, and the density at a point can be evaluated as a weighted function

of the Euclidean distance of the point to the design variable points within a specific radius. Kumar and Parthasarathy [93] proposed the use of higher order finite element through biquadratic and bicubic B-spline analysis elements, producing smoother designs compared to bilinear elements. On the other hand, Qian [94] used 2D B-spline space to characterize the density field and bilinear analysis elements and found that the degree of the B-spline field also has an effect on length-scale control. More recently, Lambe and Czekanski [95] parametrized the density field using higher order Bernstein polynomials as well as biquadratic analysis field to solve 2D mechanical problems through direct factorization of the stiffness matrix. However, due to the computational cost and higher order mesh generation, the use of higher order finite element methods has been mainly confined to 2D. To the best of my knowledge, higher order analysis and design parametrization has not been used for multimaterial thermoelastic designs. The use of higher order analysis and design allows for a more accurate evaluation of the stress field and smoother gradient compared to linear analysis, which aids in stress-constrained thermoelastic topology optimization as well as providing error control for functional-based adaptive mesh refinement.

Topology Optimization with Natural Frequency Constraints

Natural frequency constraints within topology optimization are challenging for three primary reasons (1) the issue of mode-switching, which creates non-differentiable constraints, (2) the increased multimodality of the design space, and (3) the high computational cost of solving the generalized natural frequency eigenvalue problem. Mode switching is of concern when the eigenmode associated with the fundamental natural frequency changes as the design evolves, resulting in a continuous but non-differentiable function [96]. Different works have addressed this issue by either employing a bound formulation [97, 96, 98] with mode tracking techniques [99, 100], or frequency aggregation method [101, 102, 103] which smoothly approximates the min-

imum natural frequency and creates a differentiable function. Lanczos method [44], commonly used in structural design, are computationally costly for large-scale problems since it uses direct factorization techniques. Various works have discussed a workaround by proposing different approaches. Dunning, Ovtchinnikov, Scott, and Kim [104] employed a block Jacobi conjugate gradient (BJCG) method with direct factorization to solve for a large number of buckling eigenmodes. Both Andreassen, Ferrari, Sigmund, and Diaz [105] and Federico, S., and Ole [106] solved frequency-maximization, mass-constrained topology optimization problems by replacing the eigenvalue problem with a frequency response problem, which allows efficient solution to problems with $\mathcal{O}(10^6)$ DOF. However, the frequency response representation is not suitable to act as a constraint on the natural frequency.

Contributions

This work is focused on developing new algorithms for multi-physics topology optimization and incorporating them within a framework with adaptive mesh refinement, focusing primarily on thermoelasticity. The following section, summarizes the key contributions within this thesis.

Large-scale Multimaterial Topology Optimization Framework

A large-scale multimaterial topology optimization framework was created. The framework has the capability to generate large-scale meshes directly from CAD geometry and perform multimaterial topology optimization with mesh adaption. In order to use DMO parametrizations for large-scale topology optimization, the resulting sparse partition of unity constraints must be handled efficiently. Thus, a scalable analysis and design optimization framework for multimaterial topology optimization is formulated. Within this framework, a hierarchy of analysis and design meshes are created from an input CAD geometry by local refinement of an initial coarse hexahedral

mesh. To enable this local refinement, conforming octree meshes are constructed on each element in the coarse hexahedral mesh, thereby facilitating the construction of meshes for geometric multigrid and local mesh size control around design features of interest. An Sℓ_1QP method with a trust region globalization is implemented in parallel and is tailored to handle the large number of partition of unity constraints that arise from DMO parametrizations. The adaptive refinement procedures support the capability to study whether finer meshes produce better results, revealing potential local optima. Results of this work tested the scalability of the methods and creating topology optimized multimaterial designs with as many as 330 million elements. This is further elaborated in Chapter 3

Higher Order Analysis and Design for Multimaterial Thermoelastic Topology Optimization

Building on the large-scale multimaterial topology optimization framework mentioned in the previous section, a suitable higher order thermoelastic analysis and design parametrization is incorporated. The analysis parametrization comes in the form of steady-state coupled thermoelastic analysis model and a higher order design parametrization in the form of Bernstein polynomials, which is a subset of B-spline functions. For efficient application towards large-scale designs, Bernstein polynomials has to be adapted towards geometric multigrid. An explicit formulation of the constraints on the dependent nodes as well as the interpolation across geometric multigrid levels is used. This allows the framework to be efficiently apply to the design of 3D domains. A study of different analysis and design mesh order is used to characterize its effect on design optimization. The work here is described in Chapter 4

Large-scale Topology Optimization with Natural Frequency Constraint

In order to carry out natural frequency-constrained topology optimization problems on large-scale 3D problems, the high computational cost associated with solving the eigenvalue problem at each design iteration has to be addressed. In this work, a Jacobi–Davidson method that is compatible with iterative solution techniques is used to solve the eigenvalue problem. In addition, novel eigenvector recycling strategies that reuse eigenvector information are proposed and evaluated. The combination of the Jacobi–Davidson method with the eigenvector recycling strategy enables efficient solution to large-scale topology optimization problems with natural frequency constraints. The proposed algorithms are illustrated in Chapter 5.

CHAPTER 2

INTEGRATED TOPOLOGY OPTIMIZATION FRAMEWORK

A modular framework that tightly integrates mesh generation, analysis models and optimization tool is needed to efficiently solve large-scale multi-physics topology optimization problems. For this work, the computational framework employed to obtain topology optimization designs using adaptive mesh refinement can be summarized by the schematic shown in Figure 2.1. The overall framework has three components: TMR, a geometry and meshing tool, TACS, a finite-element analysis solver, and ParOpt, an optimizer for large-scale design problems.

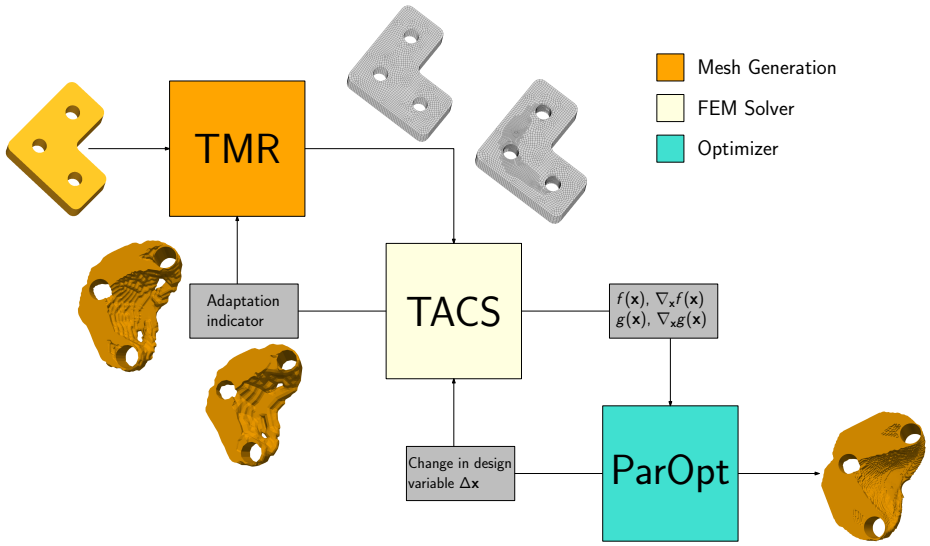


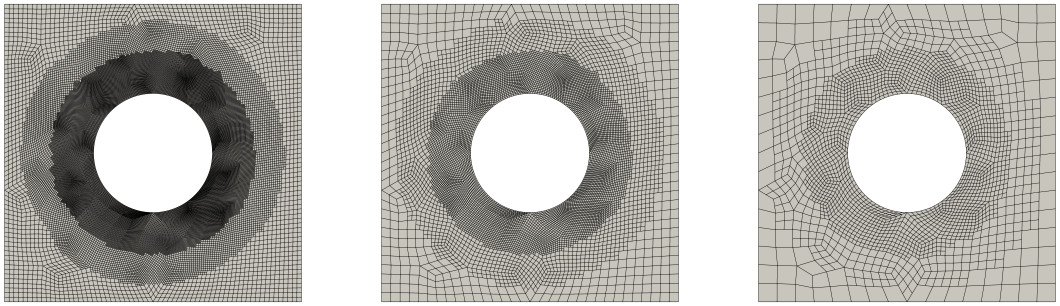
Figure 2.1: Topology Optimization Framework with Adaptive Mesh Refinement

TMR is an adaptive mesh generation tool that takes in a CAD geometry (typically in a STEP file format) and generates the mesh and its connectivity which are inputs to the structural solver TACS. TACS is an in-house finite element code which assembles and solves the governing equations for the state variables. Functions of interest $f(\mathbf{x})$, $g(\mathbf{x})$, that depend on both the design variables $\mathbf{x}$ and the state variables $\mathbf{u}$, such as

mass, compliance and stress, along with their sensitivities with respect to the design variables $\nabla_{\mathbf{x}} f(\mathbf{x})$ and $\nabla_{\mathbf{x}} g(\mathbf{x})$, are evaluated. The function values and gradients are then inputs to the optimizer ParOpt. ParOpt determines the next design point and updates the design variables accordingly. Using this design variable update, TACS can solve for the new state variables as well as recompute the functions of interest and their respective sensitivities which are fed back in ParOpt. This process is repeated until an optimized point has been found. Depending on the adaptive mesh refinement technique, the mesh can be locally refined in TMR and automatically regenerated. The corresponding finite element model can then be recreated in TACS. This process is repeated until a satisfactory design is obtained. Each of these computational components will be described in greater detail in the following sections.

Structural Solver: TACS

The structural analysis is carried out by TACS [107], which can handle large-scale design problems and structural governing equation using parallel solution algorithms. TACS can evaluate functions of interest such as mass, compliance and stress. It also evaluates the derivatives of these functions of interest with respect to the design variables efficiently using the adjoint method. At each optimization iteration, the large linear system of governing equations arising from the finite-element discretization is solved. It is not practical to solve linear system with $\mathcal{O}(10^8)$ using serial methods. Instead, a parallel geometric multigrid algorithm, which has been successfully applied to large-scale multimaterial optimization problems, is used. Other authors [108, 46] have also applied multigrid to large-scale topology optimization problems. The advantage of multigrid methods is that they offer the potential of both algorithmic scalability and parallel efficiency.



(a) Fine analysis mesh (b) Medium mesh (c) Coarsest mesh

Figure 2.2: Illustration of the construction of the meshes for geometric multigrid. The analysis mesh is the first geometric multigrid level and the subsequent geometric multigrid levels are obtained by coarsening the previous mesh through TMR.

Geometric Multigrid

This section describes the details of the geometric multigrid method implemented within the topology optimization framework. At each optimization iteration, the large linear system of governing equations arising from the finite-element discretization is solved using a parallel multigrid-preconditioned Krylov subspace method [27]. Other authors [108, 46] have also applied multigrid to large-scale topology optimization problems.

The hierarchy of meshes used within the multigrid preconditioner are generated using TMR. An illustrative example of the mesh generation procedure is shown in Figure 2.2, where the coarse mesh levels are created automatically by successive coarsening of the fine, adaptively refined analysis mesh. Each mesh level has an associated stiffness matrix, $\mathbf{K}_i(\mathbf{x}_i)$, and a solution vector $\mathbf{u}_i$ where the subscript i denotes the mesh level, ranging from $i = 1, ..., N$, where $i = 1$ is the finest mesh and $i = N$ is the coarsest mesh. In addition, $\mathbf{x}_i$ is the design vector restricted from the next finest design mesh. On all but the coarsest mesh, a block symmetric Gauss–Seidel smoothing operation is performed to damp out the high frequency solution

error. This operation is written as

$$\mathbf{u}_i \leftarrow \mathbf{G}_i(\mathbf{r}_i), \tag{2.1}$$

where $\mathbf{r}_i$ is the residual on the i-th mesh level. The restriction of the solution between mesh levels is written as

$$\mathbf{u}_{i+1} = \mathbf{R}_i \mathbf{u}_i, \tag{2.2}$$

here $\mathbf{R}_i$ is the restriction operator from the finer mesh level i to the coarser mesh level $i+1$. In this work, the interpolation operator is formed as the transpose of the restriction operator. On the coarsest mesh level, $i = N$, the system of equations $\mathbf{u}_N = \mathbf{K}_N(\mathbf{x}_N)^{-1}\mathbf{r}_N$ is solved using a parallel direct Schur-complement method [107].

Algorithm 1 Geometric multigrid algorithm with a symmetric block Gauss–Seidel smoother.

```
1:  function MULTIGRID(r_i)
2:      if i = N then
3:          return K_N(x_N)^{-1} r_N                ▷ Direct solve
4:      end if
5:      u_i ← G_i(r_i)                              ▷ Pre-smooth
6:      r_i ← r_i − K_i(x_i) u_i                    ▷ Compute the residual
7:      r_{i+1} ← R_i r_i                           ▷ Restrict the residual
8:      u_{i+1} ← MULTIGRID(r_{i+1})                ▷ Apply multigrid at the next level
9:      u_i ← u_i + R_i^T u_{i+1}                   ▷ Interpolate the solution
10:     u_i ← G_i(r_i)                              ▷ Post-smooth
11:     return u_i
12: end function
```

The complete V-cycle multigrid preconditioning algorithm is shown in Algorithm 1. The governing equations are solved using the multigrid-preconditioned GMRES(100) [109], with a maximum of two restarts, unless stated otherwise. Based on numerical experiments, I have found that this combination of multigrid preconditioning and GMRES is sufficiently robust to handle the large variability in the local element stiffness inherent to topology optimization applications.

The restriction of the design vector between mesh levels utilizes the same under-

lying operations required for multigrid. The design vector on the next coarsest design mesh is obtained as $\mathbf{x}_{i+1} = \bar{\mathbf{R}}_i \mathbf{x}_i$, where $\bar{\mathbf{R}}_i$ denotes the restriction operator for the nodal design mesh. These design restriction operations utilize the same underlying routines as the multigrid method, enabling further code reuse.

Optimizer: ParOpt

ParOpt is a parallel optimizer toolkit that can solve the large-scale design optimization problems that arise in multimaterial topology optimization applications with as many as $\mathcal{O}(10^8)$ design variables. ParOpt can handle distributed design vectors that are encountered in parallel topology optimization applications and take advantage of the sparse, linear constraints encountered in DMO multimaterial parametrization for topology optimization. ParOpt contains a number of optimization algorithms, including an interior-point method and an Sℓ_1QP method with an ℓ_∞ trust-region globalization strategy. The Sℓ_1QP method appends the inequality constraints as an ℓ_1 penalty function, using a smooth elastic-programming technique. The ℓ_∞ trust region constraint is enforced through bound constraints on the design variables. This quadratic sub-optimization problem is solved using the interior-point method in ParOpt.

Sℓ_1QP Trust Region Method

Parallel implementations of optimizers are required, since the design vector size scales with the size of the analysis problem and optimization operations would otherwise constitute a computational bottleneck. The Method of Moving Asymptotes (MMA) is commonly used to solve topology optimization problems [110], and has been parallelized for large-scale applications [46, 47]. However, topology optimization benchmark studies have demonstrated that MMA is not always the best choice, and sequential quadratic optimization or interior-point algorithms may be superior [111]. Furthermore, the inclusion of the partition of unity constraints (3.8), means that

MMA cannot be applied without modification to these problems. Thus, Paropt is designed to take advantage of the special sparsity structure of the partition of unity constraints (3.8), making it well-suited for both single material and multimaterial topology optimization.

A general formulation for mulitmaterial topology optimization problems takes the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{such that} \quad & \mathbf{A}_w\mathbf{x} = \mathbf{e} \\ & \mathbf{c}(\mathbf{x}) \geq 0 \\ & \mathbf{x} \geq 0 \end{aligned} \tag{2.3}$$

where $\mathbf{x}$ is the distributed design vector. The constraints consist of two different mathematical types. The first consist of a small number of dense constraints, $\mathbf{c}(\mathbf{x})$, whose values are duplicated on all processors. The second type consist of a large number of sparse partition of unity constraints $\mathbf{A}_w\mathbf{x} = \mathbf{e}$, that are distributed across all processors. Common multimaterial optimization problems such as mass-constrained compliance minimization and stress-constrained mass minimization problems fit within this formulation. Topology optimization with this form can be solved using an Sℓ_1QP method with a trust-region globalization [112]. Using this technique, at each iteration k, a candidate step, $\mathbf{p}_k$, is obtained by solving a non-smooth optimization problem that consists of a quadratic approximation of the objective about the point $\mathbf{x}_k$, combined with an ℓ_1 penalty function of the linearized constraint violation, leading to the following problem

$$\begin{aligned} \min_{\mathbf{p}} \quad & h_k(\mathbf{p}) = f(\mathbf{x}_k) + \mathbf{g}_k^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{B}_k\mathbf{p} + \gamma_k \left[\mathbf{c}(\mathbf{x}_k) + \mathbf{A}\mathbf{p}\right]_1 \\ \text{such that} \quad & \mathbf{A}_w\mathbf{p} = \mathbf{e} - \mathbf{A}_w\mathbf{x}_k = 0 \\ & \|\mathbf{p}\|_\infty \leq \Delta_k \\ & \mathbf{x}_k + \mathbf{p} \geq 0 \end{aligned} \tag{2.4}$$

where $[\mathbf{x}]_1 = \sum_i \max\{0, -x_i\}$, denotes the ℓ_1 norm of the constraint violation. Here, Δ_k is the trust region radius, γ_k is the penalty parameter for the ℓ_1 penalty function and $h_k(\mathbf{p})$ is the trust region model function. The linearized constraint term contains the constraint Jacobian, denoted $\mathbf{A} = \nabla_x \mathbf{c}(\mathbf{x})$. The model function, h_k, is formed from the objective gradient, $\mathbf{g}_k = \nabla_x f(\mathbf{x})$, and a quasi-Newton Hessian approximation, $\mathbf{B}_k$. Note that the problem always start from a point that is feasible with respect to the partition of unity constraints so that the constraint, $\mathbf{A}_w \mathbf{x}_k = \mathbf{e}$, is satisfied at every iteration. In this work, a quasi-Newton Hessian approximation, $\mathbf{B}_k$ is formed based on compact limited-memory BFGS updates [113]

$$\mathbf{B}_k = b_0 \mathbf{I} - \mathbf{W}_k \mathbf{M}_k \mathbf{W}_k^T, \tag{2.5}$$

where b_0 is a scalar, $\mathbf{M}_k \in \mathbb{R}^{2m \times 2m}$ is a small dense matrix and $\mathbf{W}_k$ is a matrix with $2m$ columns that is stored as a series of vectors. The approximate Hessian is updated such that it approximates the Hessian of the Lagrangian, $\mathbf{B}_k \approx \mathbf{H}(\mathbf{x}_k) \triangleq \nabla_x^2 \left(f(\mathbf{x}_k) - \mathbf{z}^T \mathbf{c}(\mathbf{x}_k) \right)$.

Algorithm 2 Sℓ_1QP method with trust region globalization.

Input: Initial design point $\mathbf{x}_1$ and initial trust region radius Δ_1
Set $k = 1$
while Optimality criteria not satisfied **do**
    Compute the candidate step $\mathbf{p}_k$ by solving (2.4)
    Evaluate the ratio $\hat{\rho}$ using Equation (2.7)
    if $\hat{\rho} \geq \eta$ **then**
        Accept step $\mathbf{p}_k$ and set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$, and $\Delta_{k+1} = \Delta_k$
    else
        if $\hat{\rho} < \hat{\rho}_l$ **then**
            Set $\Delta_k = \max(2\Delta_k, \Delta_{\min})$
        else if $\hat{\rho} > \hat{\rho}_u$ **then**
            Set $\Delta_k = \min(\Delta_k/4, \Delta_{\max})$
        end if
    end if
    Update quasi-Newton Hessian approximation
    $k \leftarrow k + 1$
end while

Once a candidate step, $\mathbf{p}_k$, is computed as a solution of the quadratic optimization problem (2.4), its acceptance is based on the merit function

$$\phi_1(\mathbf{x};\gamma_k) = f(\mathbf{x}) + \gamma_k \left[\mathbf{c}(\mathbf{x})\right]_1 , \tag{2.6}$$

and conventional trust region acceptance criteria [114]. The trust region update criteria are based on the ratio of the actual improvement in the merit function (2.6), to the improvement predicted by the model. This ratio, denoted $\hat{\rho}$, is defined as follows

$$\hat{\rho} = \frac{\phi_1(\mathbf{x}_k,\mu_k) - \phi_1(\mathbf{x}_k+\mathbf{p}_k,\mu_k)}{\phi_1(\mathbf{x}_k,\mu_k) - h_k(\mathbf{p}_k)}. \tag{2.7}$$

Based on the value of $\hat{\rho}$, a candidate step is either accepted or rejected and the model function h_k is improved through a quasi-Newton update procedure. The overall Sℓ_1QP trust region algorithm is shown in Algorithm 2. The acceptance of the step is governed by the criteria that it must make sufficient improvement such that $\hat{\rho} \geq \eta$ where the parameter takes a value of $\eta = 1/4$. The trust region radius is updated also based on the ratio $\hat{\rho}$ such that values below $\hat{\rho}_l = 1/4$ result in a decrease of the trust region radius, while values which are indicative of good progress exceeding $\hat{\rho}_u = 3/4$ result in an increase in the trust region radius.

Parallel interior-point method for convex subproblems

The step, $\mathbf{p}_k$, required in the outer Sℓ_1QP method is computed using a parallel interior-point method described in the following section. Since the quasi-Newton Hessian approximation used in the subproblem is positive definite, and the constraints are linearized, the subproblems are convex. Within each subproblem, the non-smooth ℓ_1 penalty term is reformulated using an elastic programming technique, generating an equality-constrained quadratic optimization problem with bound constraints [115, Ch 18, pg 549]. This problem always has a feasible solution, since the constraint

violation is treated through a penalty term. Throughout the remainder of this section, the iteration subscript k is dropped to simplify notation.

The efficient implementation of the parallel interior-point method leverages the special structure of the partition of unity constraints (3.8). The structure of the sparse constraint Jacobian, $\mathbf{A}_w$, is such that the matrix $\mathbf{A}_w\mathbf{D}\mathbf{A}_w^T$ is diagonal whenever the matrix $\mathbf{D}$ is a diagonal matrix. In ParOpt, the rows of the dense constraint Jacobian, $\mathbf{A}$, are stored as a series of distributed vectors, while $\mathbf{A}_w$ is accessed in a matrix-free manner through matrix-vector operations.

The interior point algorithm approximately solves a sequence of barrier problems that are designed to approach the true minimizer of the subproblem (2.4) in the limit. The barrier problem is formed by adding a interior log-penalty function to the objective to account for the bound constraints. This barrier function is designed to keep the iterates strictly in the interior of the feasible region. The barrier problem corresponding to (2.4) at design point $\mathbf{x}_k$ is

$$\begin{aligned} \min_{\mathbf{p},\mathbf{s},\mathbf{t}} \quad & \varphi(\mathbf{p},\mathbf{s},\mathbf{t};\mu) = \mathbf{g}^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{B}\mathbf{p} + \gamma\mathbf{e}^T\mathbf{t} \\ & - \mu_i \left[\log \mathbf{s} + \log \mathbf{t} + \log(\mathbf{p}-\mathbf{l}) + \log(\mathbf{u}-\mathbf{p})\right] \\ \text{such that} \quad & \mathbf{A}_w\mathbf{p} = 0 \\ & \mathbf{c} + \mathbf{A}\mathbf{p} = \mathbf{s} - \mathbf{t} \end{aligned} \tag{2.8}$$

where $\mathbf{s}$ and $\mathbf{t}$ are slack variables associated with the dense constraints. Here the components of the lower-bound vector $\mathbf{l}$ are given by $l_i = \max\{0, x_i - \Delta_k\}$, and the components of the upper bound vector $\mathbf{u}$ are given by $u_i = x_i + \Delta_k$, to conform with the variable bounds from the subproblem (2.4). The function log is the component-wise sum of the logarithm of each vector component such that $\log \mathbf{s} = \sum_i \ln s_i$. As the barrier parameter, μ_i, decreases, the minimizer of the barrier problem (2.8) approaches the KKT solution. For this work, the barrier parameter are computed by a

constant factor decrease, known as the Fiacco–McCormick approach, as follows

$$\mu_{i+1} = \sigma_i \mu_i \tag{2.9}$$

where $\sigma_i = 0.25$.

The barrier problem (2.8) is related to the perturbed KKT conditions for the optimization problem (2.3). Introducing Lagrange multipliers for the inequality cosntraints, $\mathbf{z}$, partition of unity constraints, $\mathbf{z}_w$, and the upper and lower bounds, $\mathbf{z}_l$, and $\mathbf{z}_u$, respectively, the perturbed KKT conditions can be written as follows

$$\mathbf{r} \triangleq \begin{bmatrix} \mathbf{r}_p \\ \mathbf{r}_t \\ \mathbf{r}_w \\ \mathbf{r}_z \\ \mathbf{r}_s \\ \mathbf{r}_{z_t} \\ \mathbf{r}_{z_l} \\ \mathbf{r}_{z_u} \end{bmatrix} = \begin{bmatrix} \mathbf{g} + \mathbf{B}\mathbf{p} - \mathbf{A}^T\mathbf{z} - \mathbf{A}_w\mathbf{z}_w - \mathbf{z}_l \\ \gamma\mathbf{e} - \mathbf{z} - \mathbf{z}_t \\ \mathbf{A}_w\mathbf{p} \\ \mathbf{c} + \mathbf{A}^T\mathbf{p} - \mathbf{s} + \mathbf{t} \\ \mathbf{S}\mathbf{z} - \mu_i\mathbf{e} \\ \mathbf{T}\mathbf{z}_t - \mu_i\mathbf{e} \\ (\mathbf{P} - \mathbf{L})\mathbf{z}_l - \mu_i\mathbf{e} \\ (\mathbf{U} - \mathbf{P})\mathbf{z}_u - \mu_i\mathbf{e} \end{bmatrix} = 0 \tag{2.10}$$

where $\mathbf{P} = \text{diag}\{\mathbf{p}\}$, $\mathbf{T} = \text{diag}\{\mathbf{t}\}$, $\mathbf{S} = \text{diag}\{\mathbf{s}\}$, $\mathbf{L} = \text{diag}\{\mathbf{l}\}$, and $\mathbf{U} = \text{diag}\{\mathbf{u}\}$. At each iteration of the interior-point method, an update to the design variables, slacks, and Lagrange multipliers is computed based on the solution to a linearization of the

perturbed KKT system (2.10) as follows

$$\mathbf{J}_B\Delta = \begin{bmatrix} \mathbf{B} & 0 & 0 & -\mathbf{A}_w^T & -\mathbf{A}^T & 0 & -\mathbf{I} & \mathbf{I} \\ 0 & 0 & 0 & 0 & -\mathbf{I} & -\mathbf{I} & 0 & 0 \\ \mathbf{A}_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{A} & \mathbf{I} & -\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{Z} & 0 & \mathbf{S} & 0 & 0 & 0 \\ 0 & \mathbf{Z}_t & 0 & 0 & \mathbf{T} & 0 & 0 & 0 \\ \mathbf{Z}_l & 0 & 0 & 0 & 0 & & (\mathbf{P}-\mathbf{L}) & 0 \\ \mathbf{Z}_u & 0 & 0 & 0 & 0 & & 0 & (\mathbf{U}-\mathbf{P}) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{t} \\ \Delta\mathbf{s} \\ \Delta\mathbf{z}_w \\ \Delta\mathbf{z} \\ \Delta\mathbf{z}_t \\ \Delta\mathbf{z}_l \\ \Delta\mathbf{z}_u \end{bmatrix} = -\mathbf{r}. \tag{2.11}$$

The solution of the system of equations (2.11) is the single most computationally expensive operation in the optimization algorithm. A solution for this update step, Δ, can be obtained by expressing the update equation as

$$\mathbf{J}_B\Delta = \left[\mathbf{J}_0 + \mathbf{Y}\mathbf{M}\mathbf{Y}^T\right]\Delta = -\mathbf{r}, \tag{2.12}$$

where $\mathbf{J}_0$ is a matrix obtained by replacing $\mathbf{B}$ in (2.11) with $b_0\mathbf{I}$. The matrix $\hat{\mathbf{W}}$ is

$$\hat{\mathbf{W}} = \begin{bmatrix} \mathbf{W}^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

where $\mathbf{W}$ is from the compact BFGS formula (2.5). Linear systems of the form, $\mathbf{J}_0\mathbf{y} = \mathbf{b}$, can be solved efficiently in parallel using a bordering method described in Appendix A.3.

The solution of the update equation (2.12) can be obtained using using the Sherman–Morrison–Woodbury formula [115, Ch 19, pg 597]. such that

$$\Delta = \mathbf{J}_0^{-1}\hat{\mathbf{W}}\mathbf{C}^{-1}\hat{\mathbf{W}}^T\mathbf{J}_0^{-1}\mathbf{r} - \mathbf{J}_0^{-1}\mathbf{r}$$

where the matrix $\mathbf{C} \in \mathbb{R}^{2m \times 2m}$ is given as follows:

$$\mathbf{C} \triangleq \hat{\mathbf{W}}^T \mathbf{J}_0^{-1} \hat{\mathbf{W}} - \mathbf{M}.$$

As a result, a solution of the linear system (2.12) can be obtained from the solution of $2m + 1$ linear systems of the form $\mathbf{J}_0 \mathbf{y} = \mathbf{b}$. Since the matrix $\mathbf{W}$ is stored as a series of column vectors, the operations needed to compute the solution consist primarily of dot-products of distributed vectors with a small number of matrix operations on small dense matrices. Since the vector operations parallelize efficiently, and the small matrix operations constitute a small contribution to the overall computational time, the optimization subproblem exhibits good parallel performance.

The termination criterion for each quadratic subproblem is based on the ℓ_1 norm of the perturbed KKT conditions (2.10) as follows:

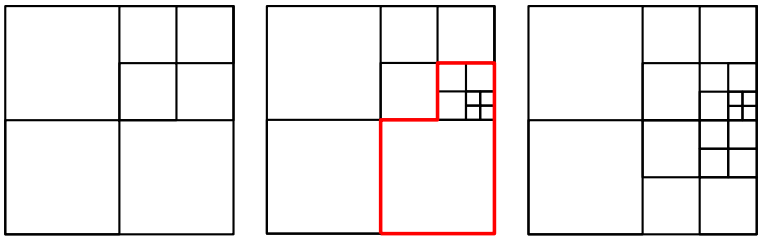
$$\max \left\{ \|\mathbf{r}_p\|_1 , \|\mathbf{r}_t\|_1 , \|\mathbf{r}_w\|_1 , \|\mathbf{r}_z\|_1 \right\} \leq \epsilon_{\text{tol}}. \tag{2.13}$$

For this work, a tight stopping tolerance of $\epsilon_{\text{tol}} = 10^{-7}$ for the subproblems is used. Based on my experience, a tight tolerance is desirable since the bound multipliers may be very small, necessitating a small value of μ_i.

Mesh Generation and Adaptivity: TMR

TMR[1] is an open source tool written in C++ that performs parallel mesh generation and adaptive mesh refinement. Within this tool, a coarse, geometry-conforming hexahedral mesh is generated from the geometry as an initial step. Next, a connected forest of octrees is created and refined on each element within the coarse hexahedral mesh, in a similar manner to the work by Burstedde, Wilcox, and Ghattas [116]. The octrees are stored using an encoding that tracks the element size and location without

[1] `https://github.com/gjkennedy/tmr`



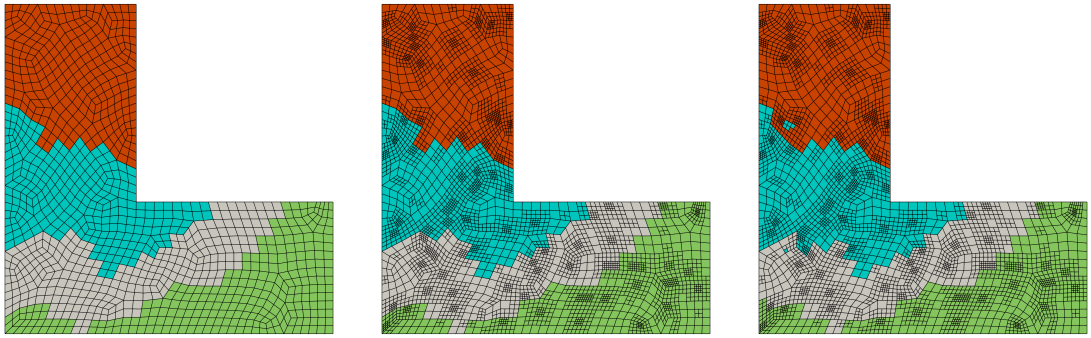
(a) Initial (b) Unbalanced (c) Re-balanced

Figure 2.3: Sequence of meshes demonstrating differences between balanced and non-balanced meshes.

storing the entire tree data structure [117, 118]. All octree operations are distributed across all processors and are performed in parallel, enabling the construction of large hierarchical meshes that are compatible with geometric multigrid. The resulting meshes must be balanced, meaning that there is no more than one level of difference in refinement across edges or faces. Figure 2.3 illustrates the process of balancing a mesh, where an initially balanced mesh, shown in Figure 2.3a, is refined, creating an non-balanced mesh shown in Figure 2.3b. Restoring the balanced property requires the adding elements in the vicinity of the refinement, shown in Figure 2.3c.

TMR uses OpenCASCADE [119] as the underlying geometry kernel, enabling the use of geometries generated from CAD. Hexahedral meshes can be either imported or generated directly using a sweeping method between source and target surfaces with the same quadrilateral mesh topology [120]. Within TMR, the source surface meshes are generated using an implementation of the Blossom-Quad algorithm [121, 122], which first generates a triangular mesh with an even number of elements, and recombines them into quadrilaterals using the Blossom perfect-matching algorithm [123, 124, 125].

To achieve good parallel scalability, the elements within the mesh must also be distributed across all processors. This is achieved by first ordering the coarse hexahderal mesh to promote a good overall hexahdral mesh partition using METIS [126]. Next, fine-grained partitioning is achieved by a repartitioning operation on the oc-



(a) Mesh distribution across 4 processors

(b) Refinement of mesh is balanced across processors

(c) Refined mesh repartitioned across processors

Figure 2.4: Illustration of balancing and repartitioning across processors.

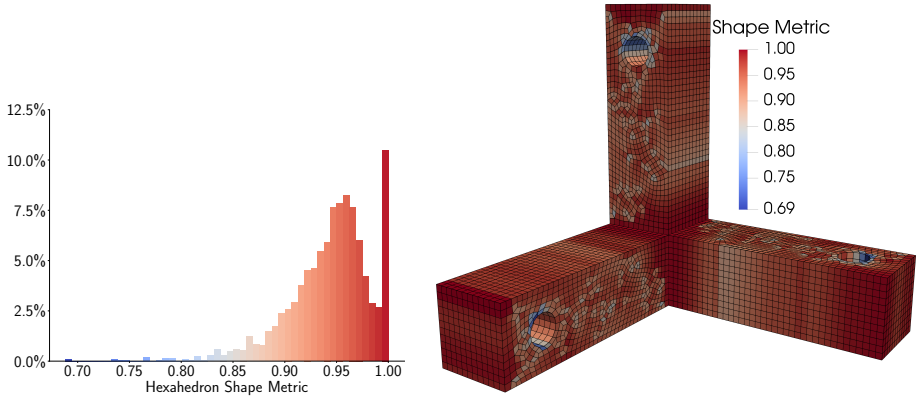
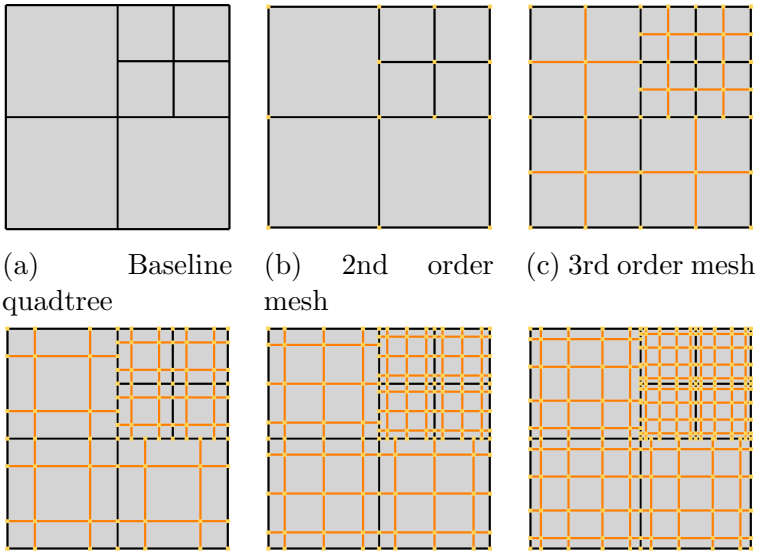


Figure 2.5: Mesh quality for the orthogonal bracket geometry.

trees such that the number of elements assigned to each processor is approximately equal. Figure 2.4a shows an L-bracket mesh that is divided across 4 processors, each denoted by a different color. When the mesh is refined, as shown in Figure 2.4b, it is automatically balanced, which changes the number of elements assigned to each processor. Figure 2.4c shows the refined mesh is repartitioned across the 4 processors to distribute the workload.

To demonstrate the capability of the mesh generation techniques in TMR, the hexahedron shape metric, first proposed by Knupp [127], is plotted to evaluate the mesh quality. The shape metric is a scale-invariant value and ranges from 0 to 1, with the value 1 denoting a cube element and 0 denoting a degenerate element. Based on CUBIT [128], an automated mesh generation toolkit used in Sandia National



(a) Baseline quadtree (b) 2nd order mesh (c) 3rd order mesh

(d) 4th order mesh (e) 5th order mesh (f) 6th order mesh

Figure 2.6: Sequence of meshes demonstrating the different mesh order that can be generated from a baseline quadtree

Laboratories, elements with the value between 0.3 and 1 are considered to be good quality. Thus this range of values for the mesh quality evaluation are adapted for this work. Figure 2.5 shows the mesh quality metric for the orthogonal bracket presented in Chapter 3.4.3. The shape metric varies between 0.68 and 1, with all elements having good quality. In all results presented here, with up to 330 million elements, the octree meshes are generated in under 20 seconds.

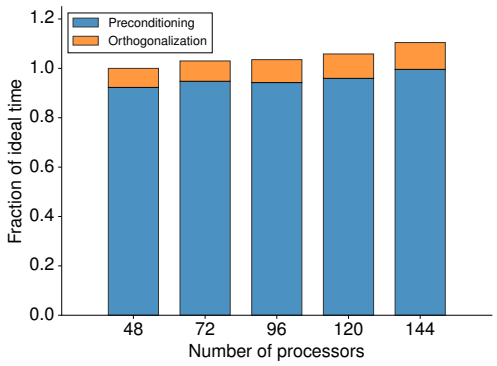
TMR also has the capability to generate higher order meshes. The creation of the higher order meshes can be built on the underlying quadtree/octree meshes. To create meshes of different orders, the quadtree has to create and assign both independent and dependent nodes to each quadrant. Figure 2.6 illustrates how different mesh orders can be generated from the same baseline quadtree, with the adaptively refined quadtree mesh outlined in black and the actual mesh and nodes generated from the quadtree outlined in orange.

From the baseline quadtree, shown in Figure 2.6a, TMR first computes the which nodes, edges and faces are owned by the quadrants on different processors. Depending

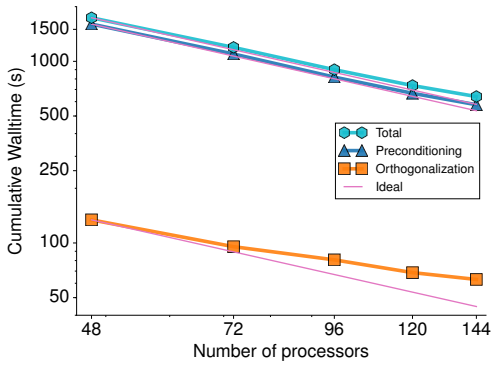
on how the quadtree is distributed across processors, the non-local quadrants that border each quadtree are passed to the corresponding processors, creating a temporary partial quadtree. TMR then creates and labels the independent and dependent nodes, depending on the mesh order, before sorting the nodes in a complete global order across all processors. Lastly, the new ordered nodes are communicated locally across quadrants and returned to the same quadtrees that are bordered by non-local quadrants. The resulting 2nd and 3rd order meshes are shown in Figures 2.6b and 2.6c. Due to the numerical instability effects (Runge phenomenon) for higher order interpolation on uniform grids, for this work, Gauss-Lobatto points are used to generate higher order meshes as shown in Figures 2.6d, 2.6e, and 2.6f. Generating hexahedral meshes with different mesh orders from octrees are analogous to that of quadtrees.

Scalability Results

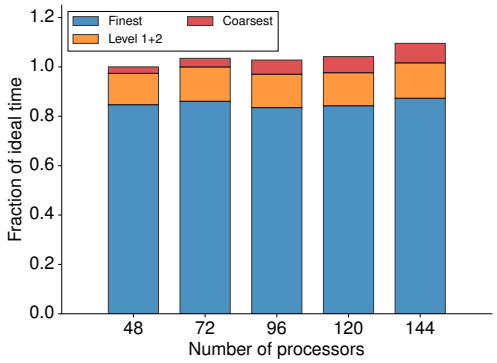
The scalability of the algorithms are first demonstrated on a topology optimization problem with 18.9 million elements, 19.2 million nodes, and 7.32 million design variables. For this study, the number of optimization iterations is fixed at 50 and the number of processors are varied to investigate the computational efficiency of individual components of the framework. The elements are distributed equally across all processors on all geometric multigrid levels except for the coarsest level where they are divided among 24 processors. Figure 2.7 shows the fraction of the ideal cost of each optimization component as a function of the number of processors. Figure 2.7a and 2.7c show the computational efficiency of GMRES and the geometric multigrid preconditioner, respectively. These components of the framework scale well from 48 to 144 processors where the multigrid preconditioner scales up to 144 processors with 90.8% efficiency, and GMRES overall scales with 90.5% efficiency. From Figure 2.7a and 2.7b, it is observed that the small inefficiency arises from the increase in the



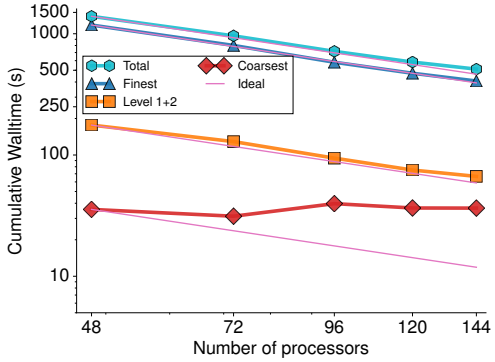
(a) Solution Efficiency



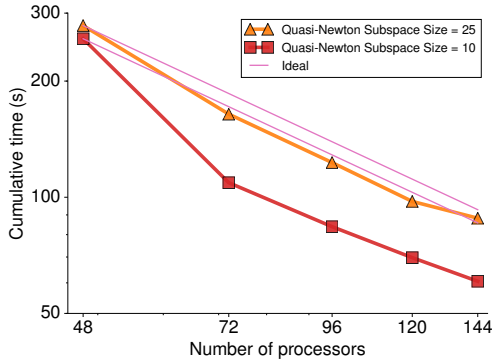
(b) Cumulative Solution Cost



(c) Geometric Multigrid Efficiency



(d) Cumulative Geometric Multigrid Cost



(e) Cumulative ParOpt Cost

Figure 2.7: Scalability of the individual framework components as a function of the number of processors.

fraction of time spent in the orthogonalization step of the solution. Figure 2.7c shows that the overall geometric multigrid cost scales close to ideally as the number of processors increase. Figure 2.7d shows that the fraction of time taken on the coarsest level increases for increasing number of processors due to the use of the direct solver. The slight increase in computational time for the direct solver is due to additional communication costs since the computation at this level is restricted to 24 processors.

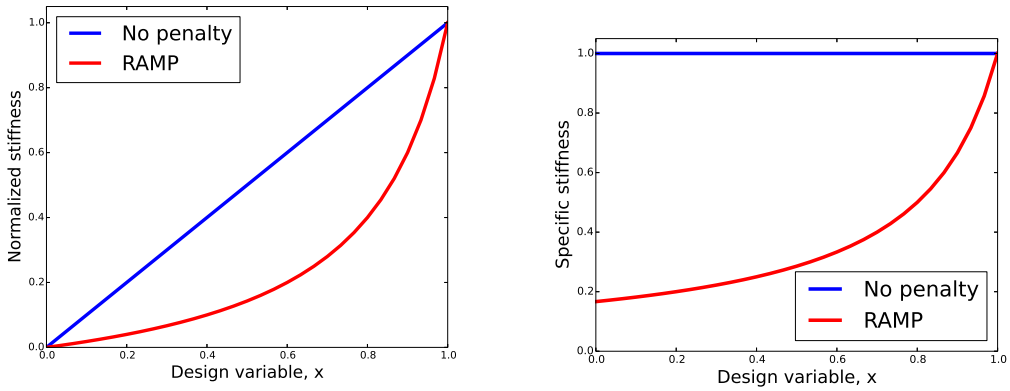
Figure 2.7e shows the computational cost of the ParOpt optimizer for different quasi-Newton subspace sizes. The computational time increases with subspace size due to the additional operations required for additional set of quasi-Newton vectors. The optimization steps scale at better than the ideal rate between 48 and 72 processors, which can be attributed to superior cache efficiency. The optimization steps scale at a near ideal rate between 72 to 144 processors. Note that the computational cost of the optimizer is no more than 20% of the computational cost of the solution method.

CHAPTER 3
COUPLED STEADY-STATE THERMOELASTIC ANALYSIS AND MULTIMATERIAL TOPOLOGY OPTIMIZATION

In this work, high-fidelity multi-physics models are needed for topology optimization. Towards this goal, the development of a coupled steady-state thermoelastic analysis formulation for multimaterial topology optimization is described in this chapter. First the design parametrization of topology optimization is illustrated. Then, the steady-state coupled thermoelastic analysis is formulated to include the topology optimization design parametrization. Lastly, the adaptive mesh refinement heuristics used is described and the resulting designs are presented.

Single and Multimaterial Parametrization

For general single-material topology optimization problems, the domain is discretized into N elements that can either be void or solid, corresponding to design variable, x_j, of 0 or 1. However, instead of solving a combinatorial problem, the binary requirement on the design variable, x_j, is relaxed, allowing it to lie continuously within the range $x_j \in [\epsilon,\ 1]$, where $\epsilon > 0$ is a small but finite value that denotes the lower bound. To obtain a nearly discrete design, a stiffness penalization is used. The penalization is described by either the Solid Isotropic Material with Penalization (SIMP) approach [1] or the Rational Approximation of Material Properties (RAMP) approach [129]. Figure 3.1 shows the effect of using penalization on the normalized and specific stiffness of the domain. For both SIMP and RAMP schemes, the intermediate stiffness to mass ratio is penalized relative to the design at the upper or lower bound, making such designs less favorable compared to one that consists of mostly solid or void elements. In both instances, the stiffness interpolation is a nonlinear convex function takes the



(a) Normalized stiffness of element as a function of design variables

(b) Specific stiffness of element as a function of design variables

Figure 3.1: Illustration of the mechanism of penalization that creates a nearly discrete design

form

$$E_j(\mathbf{x}) = w_j(\mathbf{x})E, \tag{3.1}$$

where $w_j(\mathbf{x})$ is a weighting function, while the density interpolation is linear

$$\rho_j(\mathbf{x}) = x_j\rho. \tag{3.2}$$

Here $\mathbf{x} \in \mathbb{R}^N$ are the design variables, with N variables in total, E is the Young's modulus and ρ is the density of the solid material. The scheme-dependent weight function, $w_j(\mathbf{x})$, is

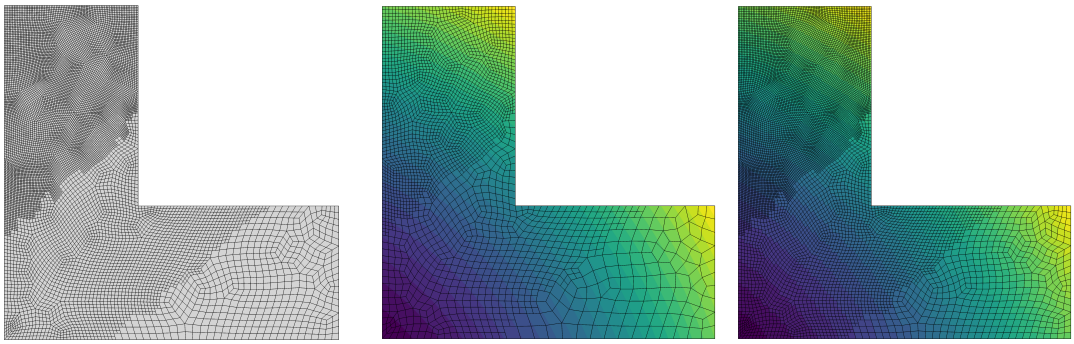
$$w_j(\mathbf{x}) = x_j^p, \tag{3.3}$$

for the SIMP method, where p is the SIMP exponent, and

$$w_j(\mathbf{x}) = \frac{x_j}{1 + q(1 - x_j)}, \tag{3.4}$$

for the RAMP method, where q is the RAMP parameter.

For this work, in order to achieve an efficient and consistent design representation



(a) Analysis mesh (b) Design mesh (c) Interpolated design variables

Figure 3.2: Illustration of the construction of the design parametrization. The analysis mesh is coarsened to create the design mesh. Design variables are defined at the nodes. Density values are interpolated back to the elements on the analysis mesh.

between meshes with different levels of local refinement, a node-based parametrization [130], where the design field is evaluated on a coarsened copy of the analysis mesh, is used. This node-based parametrization is well-suited for the geometric multigrid solution strategies that are described in Section 2.1.1. Figure 3.2 illustrates the process whereby the initial analysis mesh is coarsened, and the design density associated with each element in the finer analysis mesh is interpolated from the coarse design mesh [90]. This linear interpolation process is written as

$$\xi = \mathbf{F}\mathbf{x} \tag{3.5}$$

where $\mathbf{x}$ are the design variables on the coarser design mesh, ξ are the interpolated design variables on the finer analysis mesh and $\mathbf{F}$ is the matrix that interpolates the design variables across the coarser design mesh to the finer analysis mesh. Note that a bilinear/trilinear interpolation is employed, so the rows of $\mathbf{F}$ satisfy a partition of unity property. Using this parametrization, the stiffness for each element j in the analysis mesh can be computed from the interpolated values

$$\mathbf{D}(\mathbf{x})_j = \left(\frac{\xi}{1 + q_D(1-\xi)} + \kappa_0\right)\mathbf{D}_j. \tag{3.6}$$

where $\kappa_0 = 10^{-6}$ is a small constant term added to ensure that the stiffness matrix is non-singular. The density of the element is also evaluated from the interpolated variables

$$\rho_j(\mathbf{x}) = \xi\rho_j.$$

Extending this parametrization to multimaterial design, each node in the design mesh has $M + 1$ design variables, consisting of one topology optimization variable, x_1, associated with a void material, and M candidate material selection variables, x_i, for $i = 2, \ldots, M + 1$. For each node, the void or candidate material is selected if $x_j = 1$, while $x_i = 0$ for $i \neq j$. After the interpolation step (3.5), each element has $M + 1$ interpolated material values, corresponding to the interpolated topology variable and candidate material selection variables. For the stiffness properties, a DMO-type parametrization [19, 24] that takes the following form for each element in the analysis mesh is used

$$\mathbf{D}_j(\mathbf{x}) = \sum_{i=1}^{M} \left(\frac{\xi_{i+1}}{1 + q_D(1 - \xi_{i+1})} + \kappa_0 \right) \mathbf{D}_{i,j}. \tag{3.7}$$

where similar to the single material case, $\kappa_0 = 10^{-6}/M$ is a small constant term added to ensure that the stiffness matrix is non-singular. The density of the element is also determined from the interpolated variables

$$\rho(\mathbf{x}) = \sum_{i=1}^{M} \xi_{i+1}\rho_{i,j}.$$

Further details of this method can be found in [24]. To ensure a reasonable interpolation of intermediate designs, DMO methods impose a partition of unity constraint on the interpolated element variables, $\boldsymbol{\xi}$, used in the stiffness interpolation (3.7). However, this constraint can be imposed directly on the nodal variables, due to the nature

of the partition of unity interpolation, such that

$$\sum_{i=1}^{M+1} x_i = 1, \tag{3.8}$$

for every node in the design mesh. Therefore, the partition of unity constraint (3.8) results in many sparse, linear constraints: one for each node in the design mesh. Note that the constraint (3.8), automatically imposes that each design variable will be bounded from above by unity, so the upper design limit can be omitted.

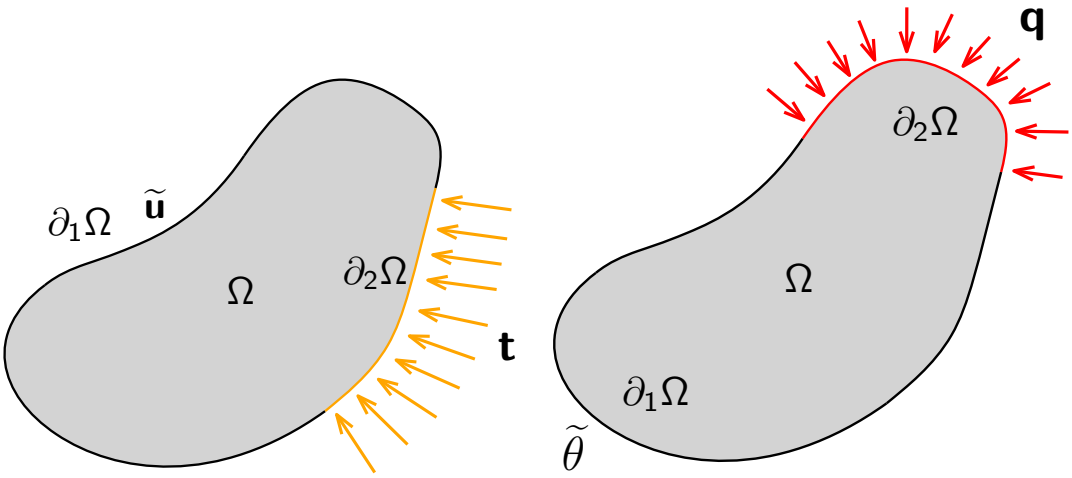
Steady-state Coupled Thermoelastic Formulation

The expressions given in Section 3.1 are sufficient for problems with only static mechanical loads but insufficient with design-dependent thermal loading. A general thermoelastic heat transfer formulation accounts for the interaction between the heat transfer problem and the structural mechanics. for this work, temperature-dependent material properties are not considered, so only a one-way coupling formulation is employed. The structural problem and heat transfer problem can each be represented by a different PDE, which are then coupled through the strain due to thermal expansion. The thermal and elastic problems are illustrated in Figure 3.3.

The governing equations for elasticity are given by

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0, \quad & \text{on } \Omega \\ \mathbf{u} = \widetilde{\mathbf{u}} \quad & \text{on } \partial_1 \Omega \\ \sigma \cdot \mathbf{n} = \mathbf{t} \quad & \text{on } \partial_2 \Omega, \end{aligned} \tag{3.9}$$

where $\mathbf{b}$ is the body force, $\mathbf{n}$ is the unit normal to the surface $\partial_2\Omega$, $\mathbf{t}$ is the traction applied to $\partial_2\Omega$ and $\widetilde{\mathbf{u}}$ is the prescribed displacement on $\partial_1\Omega$. For this work, the body force is zero. The constitutive model employed in this work is linear, resulting in the



(a) Domain under structural loads and boundary conditions

(b) Domain under thermal loads and boundary conditions

Figure 3.3: A general domain under combined structural and thermal loading

relationship

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{D}} \cdot (\nabla_s \mathbf{u} - \boldsymbol{\alpha}\theta),$$

where $\boldsymbol{\epsilon} = \nabla_s \mathbf{u} = \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u})$ is the strain, and $\boldsymbol{\alpha}$ are the thermal coefficients of expansion, and $\boldsymbol{\mathcal{D}}$ is the constitutive tensor.

The governing differential equation for the heat transfer problem, derived from the first law of thermodynamics (3.10) is

$$\begin{aligned} \nabla \cdot (\kappa \nabla \cdot \theta) + \mathbf{Q} = 0, \quad & \text{on } \Omega \\ \theta = \widetilde{\theta} \quad & \text{on } \partial_1\Omega \\ (\kappa \nabla \cdot \theta) \cdot \mathbf{n} = \mathbf{q} \quad & \text{on } \partial_2\Omega, \end{aligned} \tag{3.10}$$

where κ is the thermal conductivity of the material, $\mathbf{Q}$ is the volumetric heating source, $\mathbf{n}$ is the unit normal to the surface $\partial_2\Omega$, $\mathbf{q}$ is the heat flux at the surface and $\widetilde{\theta}$ is the prescribed change in temperature θ on $\partial_1\Omega$. Both (3.9) and (3.10) must be solved simultaneously to obtain both the displacement $\mathbf{u}$ and the change in temperature θ for the domain [131]. For a steady-state problem, both equations can be simplified to one-way coupling between the heat transfer equations and the equations of linear

elasticity. With no general analytical solution available, it is necessary to introduce the finite element form of these equations as well as the concept of generalized thermal stress coefficient (GTSC) in as originally described in Gao and Zhang [132] and Gao, Xu, and Zhang [25].

Assuming that the thermal expansion coefficient of the material $\alpha(\mathbf{x})$ is independent of temperature and the domain is in a steady-state temperature field, the effects of design-dependent thermal load can be written as

$$\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}^{\mathrm{m}} + \mathbf{f}^{\mathrm{th}}(\mathbf{x}) \tag{3.11}$$

where $\mathbf{u}$ are the structural displacements, $\mathbf{f}^{\mathrm{m}}$ are the externally applied mechanical loads and $\mathbf{f}^{\mathrm{th}}(\mathbf{x})$ are the design-dependent thermal loads. For each element, j, the design-dependent thermal loads, $\mathbf{f}_j^{\mathrm{th}}$, can be expressed as

$$\mathbf{f}_j^{\mathrm{th}} = \int_{\Omega_j} \mathbf{B}_j^{\mathrm{T}} \mathbf{D}_j \epsilon_j^{\mathrm{th}} d\Omega, \tag{3.12}$$

where Ω_j denotes the element domain and $\mathbf{B}_j$ is the element-wise strain-displacement matrix that is independent of design variable x_j. The element-wise constitutive matrix, $\mathbf{D}_j$, takes the form of (3.6) for single material or (3.7) for multimaterial. The thermal strain, ϵ_j^{th}, is defined as

$$\epsilon_j^{\mathrm{th}} = \alpha_j(\mathbf{x})(\theta_j)\phi^{\mathrm{T}}, \tag{3.13}$$

where θ_j is the change in temperature of the j-th element with respect to a reference temperature. The vector ϕ is defined as $\phi = [\,1\ 1\ 1\ 0\ 0\ 0\,]$, for 3D problems, and $\phi = [\,1\ 1\ 0\,]$, for 2D problems.

Using the definition of the thermal strain (3.13), the thermal load expression (3.12)

yields

$$\mathbf{f}_j^{\text{th}} = \theta_j \int_{\Omega_j} \mathbf{B}_j^{\mathrm{T}} \mathbf{D}_j(\mathbf{x}) \alpha_j(\mathbf{x}) \phi^{\mathrm{T}} d\Omega, \tag{3.14}$$

where the assumption is that the penalized stiffness, coefficient of thermal expansion and temperature are constant over each element. Since only the product $\mathbf{D}_j(\mathbf{x})\alpha_j(\mathbf{x})$ in Equation (3.14) depends on the design variables, the thermal stress coefficient (TSC), for single material design, or the generalized thermal stress coefficient (GTSC), for multimaterial design can be defined as

$$\beta_j(\mathbf{x}) = \mathbf{D}_j(\mathbf{x})\alpha_j(\mathbf{x}), \tag{3.15}$$

and $\beta_j(\mathbf{x})$ can be treated as an inherent material property. To solve a problem where the thermal loading is nonuniform in the domain, the formulation (3.11) is insufficient and requires solution to the heat transfer problem as well. The coupled thermoelastic heat transfer problem exhibits a one-way coupling between the heat transfer equations and the equations of linear elasticity as follows

$$\begin{bmatrix} \mathbf{K}(\mathbf{x}) & \mathbf{L}(\mathbf{x}) \\ 0 & \mathbf{H}(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix}, \tag{3.16}$$

Here, $\mathbf{K}(\mathbf{x})$ is the stiffness matrix associated with the structural problem, $\mathbf{H}(\mathbf{x})$ is the thermal conductivity matrix associated with the heat transfer problem, and $\mathbf{L}(\mathbf{x})$ is the thermoelastic coupling matrix that accounts for the body-forces produced by the temperature-induced volumetric change. The right-hand-side of (3.16) consists of the loads associated with the structural problem, $\mathbf{f}$, and the heat transfer problem, $\mathbf{q}$, respectively.

For a complete parametrization of this coupled system for topology optimization, additional material parameters must be defined as a function of the element material selection variables in an analogous manner to the stiffness interpolation (3.6) or (3.7).

These additional parameters consist of the heat conductivity coefficient $\lambda(\mathbf{x})$, the coefficient of thermal expansion, $\alpha(\mathbf{x})$ which consequently form the GTSC denoted as $\beta(\mathbf{x})$. A DMO-type interpolation, as employed in the previous works [84], are used for each of these coefficients, denoting the penalties in these cases as q_λ and q_β, respectively. Using these additional material property interpolations, the thermal conductivity matrix, $\mathbf{H}(\mathbf{x})$ can be evaluated as

$$\mathbf{H}(\mathbf{x}) = \int_\Omega \mathbf{B}^{\mathrm{th}^{\mathrm{T}}} \lambda(\mathbf{x}) \mathbf{B}^{\mathrm{th}} d\Omega$$

where $\mathbf{B}^{\mathrm{th}}$ defines the relationship between the change in temperature, θ and the spatial gradient of the change in temperature. Lastly, the coupling matrix $\mathbf{L}(\mathbf{x})$ in the linear system (3.16), can be expressed as

$$\mathbf{L}(\mathbf{x}) = -\int_\Omega \mathbf{B}^{\mathrm{T}} \mathbf{D} \alpha(\mathbf{x}) \phi^{\mathrm{T}} \mathbf{N}^T d\Omega,$$

where $\mathbf{N}$ are the shape functions and $\mathbf{L}(\mathbf{x})$ incorporates the design-dependent thermal load. When the change in temperature field is uniform, the coupled system of equations can be simplified to give

$$\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} - \mathbf{L}(\mathbf{x})\,\theta. \tag{3.17}$$

where $\mathbf{f}^{\mathrm{th}}(\mathbf{x}) = -\mathbf{L}(\mathbf{x})\,\theta$ as shown in (3.11). Similar to the structural problem where there are tractions and body forces, the corresponding loads in the heat transfer problem are heat flow and heat sink/source.

Stress Recovery and Constraint Evaluation

Part of this work includes imposing stress constraints on coupled thermoelastic problems with multiple materials. To develop these stress constraints, unlike the case of

single material, an equivalent stress for each material selection in each element has to be formulated. This equivalent stress is then used to evaluate a failure criteria for each material, which is subsequently used to evaluate an element-wise failure index. To make the stress-constrained problem computationally tractable, the element-wise failure index is aggregated into a single global stress constraint using a Kreisselmeier–Steinhauser (KS) functional.

For a single material, the local stress can be written as

$$\sigma = \mathbf{D}\left(\mathbf{B}\,\mathbf{u} - \alpha\,\phi\,\mathbf{N}\,\theta\right) \tag{3.18}$$

where $\mathbf{D}$ and α are the constitutive matrix and the coefficient of thermal expansion, respectively for a single material. Note that the displacement vector depends implicitly on the design variables through the governing equations. To incorporate stress constraints into a multimaterial thermoelastic design problem, this expression for the stress (3.18) is used for each material to obtain the solid stress

$$\sigma_i = \mathbf{D}_i\left(\mathbf{B}\,\mathbf{u} - \alpha_i\,\phi\,\mathbf{N}\,\theta\right). \tag{3.19}$$

Note that this expression uses the material constitutive matrix, $\mathbf{D}_i$ and thermal coefficient of expansion, α_i, without penalization. To compute whether the i-th candidate material has failed, a failure criterion for each candidate is evaluated, written as $F_i(\sigma_i)$, where a value of $F_i(\sigma_i) > 1$ would indicate material failure if $\xi_{i+1} = 1$. In this work, the von-Mises failure criterion is used for all materials. To combine these failure index values into a single failure value, the effective failure index can be evaluated as

$$F(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{M} w_i(\mathbf{x}) F_i(\sigma_i),$$

where $w_i(\mathbf{x})$ are design-dependent stress-relaxation weights. These design-dependent

weighting functions are similar to other stress relaxation functions used for failure evaluation in the context of topology optimization [75, 23, 133]. For this work, I use an interpolation that follows Le, Norato, Bruns, Ha, and Tortorelli [133], with

$$w_i(\mathbf{x}) = \frac{\xi_{i+1}}{\epsilon(1-\xi_{i+1})+\xi_{i+1}},$$

where ϵ takes the value of 0.2.

Using a global stress aggregation technique,the element-wise failure index from all elements in the mesh can be condensed into a single function. The continuous form of the Kreisselmeier-Steinhauser (KS) aggregation, as described by Kennedy and Hicken [134], is used and formulated as

$$c_{\mathrm{KS}}(\mathbf{x},\mathbf{u};\rho_{\mathrm{KS}}) = m + \frac{1}{\rho_{\mathrm{KS}}}\ln\left[\int_\Omega e^{(F-m)}d\Omega\right],$$

where $m \triangleq \max F(\mathbf{x},\mathbf{u})$ is the maximum value of the equivalent failure index over the entire domain. While the continuous form of the KS aggregation is non conservative, it is preferred over the discrete form of the KS aggregation since it exhibits mesh-independent convergence properties. This property is especially important when the aggregation technique is used in conjunction with adaptive mesh refinement where it is desirable for the aggregation to be consistent across adaptively refined mesh.

Sensitivity Evaluation

To utilize gradient-based optimization techniques, the sensitivity of the functions must be evaluated efficiently. For the work shown here, the adjoint method is used. For ease of derivation, the coupled thermoelastic heat transfer problem (3.16) is represented by

$$\mathbf{A}(\mathbf{x})\mathbf{v} = \mathbf{b}, \tag{3.20}$$

The Lagrangian can be formed as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^{\boldsymbol{T}} \left[\mathbf{A}(\mathbf{x})\mathbf{v} - \mathbf{b}\right], \tag{3.21}$$

To evaluate the sensitivity of $f(\mathbf{x})$, (3.21) is differentiated.

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})}{\mathrm{d}\mathbf{x}} &= \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + \frac{\partial f(\mathbf{x})}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \boldsymbol{\lambda}^{\boldsymbol{T}}\left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}} + \frac{\partial \mathbf{A}}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right] \\ &= \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{v}} + \boldsymbol{\lambda}^{\boldsymbol{T}}\frac{\boldsymbol{\partial} \mathbf{A}}{\boldsymbol{\partial} \mathbf{v}}\right)\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \boldsymbol{\lambda}^{\boldsymbol{T}}\frac{\partial \mathbf{A}}{\partial \mathbf{x}} \end{aligned} \tag{3.22}$$

To use the adjoint method, the adjoint vector $\boldsymbol{\lambda}$ can be evaluated as

$$\boldsymbol{\lambda} = -\left(\frac{\partial \mathbf{A}}{\partial \mathbf{v}}\right)^{-\boldsymbol{T}} \frac{\partial f(\mathbf{x})}{\partial \mathbf{v}}$$

Substituting the adjoint vector $\boldsymbol{\lambda}$ into (3.22), the total derivative of $f(\mathbf{x})$ with respect to design variables $\mathbf{x}$

$$\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + \left[-\left(\frac{\partial \mathbf{A}}{\partial \mathbf{v}}\right)^{-\boldsymbol{T}} \frac{\partial f(\mathbf{x})}{\partial \mathbf{v}}\right]^{\boldsymbol{T}} \frac{\partial \mathbf{A}}{\partial \mathbf{x}} \tag{3.23}$$

Adaptive Mesh Refinement Heuristic

As discussed in Section 1.2, there are different adaptive mesh refinement indicators that have been employed for topology optimization such feature-based methods [55], solution-based methods [61], and functional output-based methods [64]. For this work, feature-based refinement is demonstrated. In the feature-based adaptive refinement approach, elements are refined or coarsened based on the optimized material distribution. Here I use a simple heuristic where the mesh is refined in locations where the optimizer has placed material, and coarsened in locations where the optimizer has allocated the void material. This is applicable to both single material and mul-

timaterial design problems, with slight differences in variable convention. For single material design, the parameter x_{rl} defines the lower limit on the solid material, below which the elements are coarsened $\xi \leq x_{rl}$. The parameter x_{ru} defines a upper limit on the solid material, above which the element is refined. For single-material design, I use values of $x_{rl} = 0.05$ and $x_{ru} = 0.25$, unless stated otherwise. For multimaterial design, the parameters are now defined for the void material selection variable ξ_1, which has an reciprocal relationship with the solid parameter for single material design. x_{rl} defines an upper limit on the void material, above which elements with void material selection variables are coarsened whenever $\xi_1 \geq x_{rl}$, while x_{ru} defines a lower limit on the void material, below which the element is refined. For multimaterial design, the values of x_{rl} and x_{ru} are chosen to be $x_{rl} = 0.95$ and $x_{ru} = 0.75$ respectively, unless stated otherwise. The value of x_{rl} is chosen such that there is almost no solid material in the region where the mesh is coarsened and the value of x_{ru} is chosen so that it refines the mesh in areas where there is either intermediate or converged material. After the mesh adaptation step is invoked, the design on the new mesh is interpolated from the previously optimized design using TMR. The overall feature-based adaptive refinement heuristics used in this work shown in Algorithm 3.

Algorithm 3 Topology optimization with adaptive mesh refinement

1: **Input:** CAD geometry; initial mesh spacing
2: Generate initial analysis and design meshes in TMR
3: Optimize to obtain initial topology
4: **while** Termination criterion not satisfied **do**
5: Identify elements to coarsen with filtered density
6: Identify elements to refine with filtered density
7: Regenerate the mesh in TMR based on updated refinement
8: Interpolate design variables onto the new mesh
9: Restart the topology optimization
10: **end while**

Results

Two different types of topology optimization problems are tackled in this work: mass-constrained compliance-minimization, and stress-constrained mass-minimization for both single and multimaterial design. The mass-constrained, compliance-minimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \mathbf{u}^{\mathbf{T}}\mathbf{K}(\mathbf{x})\mathbf{u} \\ \text{such that} \quad & \mathbf{x} \geq 0 \\ & \mathbf{A}_w\mathbf{x} = \mathbf{e} \\ & \mathbf{c}(x) = m_{\text{fixed}} - m(\mathbf{x}) \geq 0 \\ \text{governed by} \quad & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} - \mathbf{L}(\mathbf{x})\,\theta \end{aligned} \tag{3.24}$$

where $\mathbf{x}$ are the design variables, $f(\mathbf{x})$ is the compliance of the structure, $\mathbf{u}$ are the displacement state variables, $\mathbf{f}$ is the force vector, and $\mathbf{A}_w\mathbf{x} = \mathbf{e}$ is the partition of unity constraint (3.8) that is included for multimaterial design. The stress-constrained mass-minimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & m(\mathbf{x}) \\ \text{such that} \quad & \mathbf{x} \geq 0 \\ & \mathbf{A}_w\mathbf{x} = \mathbf{e} \\ & c_{\text{KS}}(\mathbf{x}, \mathbf{u}; \rho_{\text{KS}}) \leq c_u \\ \text{governed by} \quad & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} - \mathbf{L}(\mathbf{x})\,\theta \end{aligned} \tag{3.25}$$

where $m(\mathbf{x})$ is the mass of the structure, $\mathbf{u}$ are the displacement state variables, and $\mathbf{x}$ are the design variables. The stress constraint $c_{\text{KS}}(\mathbf{x}, \mathbf{u}; \rho_{\text{KS}})$ is as described in the previous section and c_u is the upper bound on the ratio of the KS value and the failure stress. In this work, the methods employed are presented in [134] and [135].

The results obtained in subsequent sections were run on a dedicated cluster within the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology. The cluster has 20, 2.50GHz Intel Xeon CPU E5-2680-v3 compute nodes. Each node has 24 processor cores with a total of 128 GB of RAM per node.

2D Thermoelastic Single Material Topology Optimization Results

In this section, common benchmark problems that have been used in other works for thermoelastic topology optimization are illustrated and the corresponding results obtained from using adaptive mesh refinement are presented. While it is noted in Section 1.3 that the stress-based objective produces more realistic design, both the compliance-based and stress-based designs are obtained using adaptive mesh refinement to compare against other authors work in the literature. For both design problems, identical initial mesh sizes are used. To illustrate the difference in the structural layout for the two problems with the same mass, results from the stress-constrained mass-minimization problem are used as the corresponding mass for the mass-constrained compliance-minimization problem. In these results, penalization parameters are set to values of $q_D = 8.0$, $q_\lambda = 0.0$, and $q_\beta = 8.0$, respectively.

The bi-clamped beam problem shown in Figure 3.4 was first introduced for thermoelastic topology optimization by Rodrigues and Fernandes [65]. For this problem, the plate is taken to be 1 cm thick, clamped at both ends with a value of L = 0.04 m. The load P is modeled as a traction with magnitude 300 MPa, along a 5 mm edge which has its two endpoints equidistant from the midpoint of the lower edge. This is so that stress concentration is minimized. The uniform change in temperature field θ is set at 15°C. The material chosen for this problem is 4340 steel and its properties are listed in Table 3.1.

The domain is discretized into an initial mesh size of 44 by 32 elements, for a total

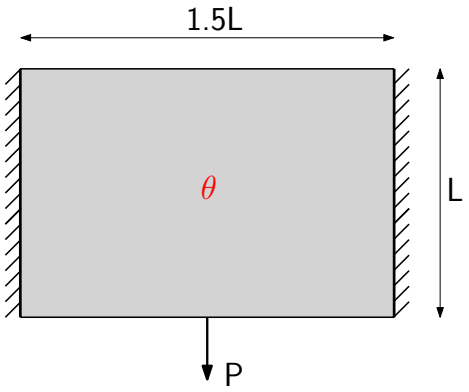


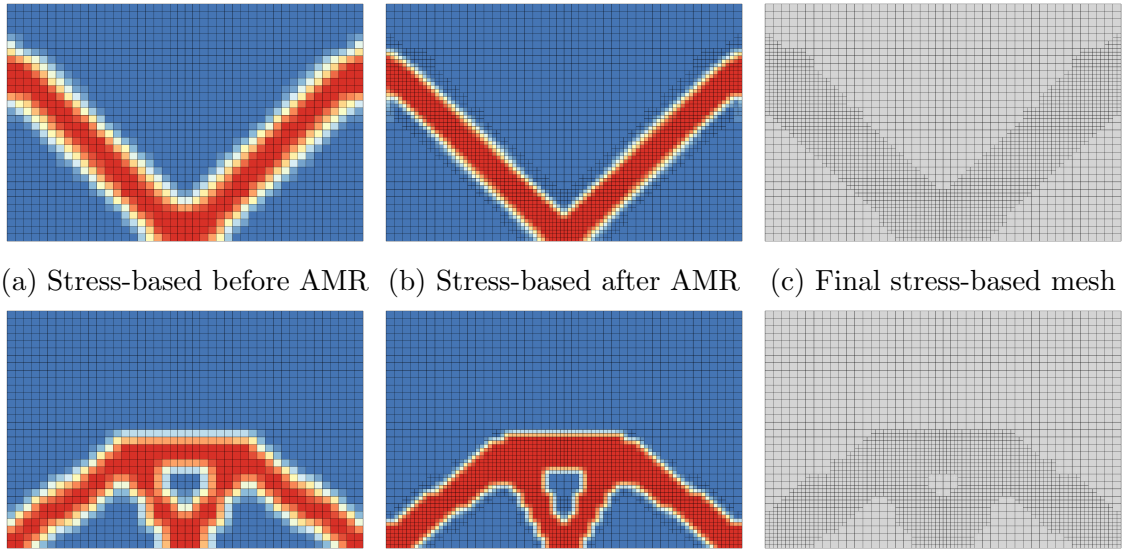
Figure 3.4: Bi-clamped domain with load P and uniform change in temperature field

Table 3.1: Material properties of 4340 Steel

Material Property	Value
E (GPa)	210.0
Poisson's ratio ν	0.3
Yield strength (MPa)	400.0
Coefficient of Thermal Expansion α (10^{-6}/°C)	11.0
Thermal Conductivity (W/(m°C))	44.5

of 1408 elements. A uniform initial design variable is set at $\mathbf{x} = 0.95$, and adaptive mesh refinement is carried out once after the design converged on the initial mesh. For the compliance-minimization problem, the volume constraint is set at 20%, which is the volume obtained from stress-constrained mass-minimization problem. Figure 3.5 shows the resulting design with solid denoted in red and void in blue.

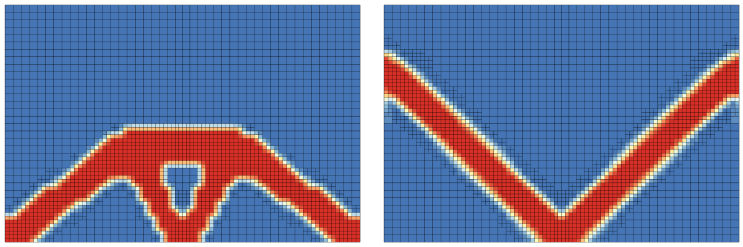
From Figures 3.5e and 3.5b, similar to previous work [75], compliance-based and stress-based designs produced vastly different layout even for the same mass, with the compliance-based design having more of the structure sustaining compression compared to the stress-based design. Also, note that with adaptive mesh refinement, designs with smoother boundaries and minimal intermediate densities are obtained. Figure 3.6 compares the difference in structural layout with and without thermal loading. It is observed that the compliance-based and stress-based designs with only mechanical loads do not exhibit the drastic differences that is evident with the compliance-based and stress-based designs that has combined mechanical and



(a) Stress-based before AMR (b) Stress-based after AMR (c) Final stress-based mesh

(d) Compliance before AMR (e) Compliance after AMR (f) Final compliance mesh

Figure 3.5: Comparison of compliance-based and stress-based bi-clamped plate designs with adaptive mesh refinement. The first stress based result has a volume of m = 0.22, and the final stress-based design has a volume of m = 0.2. The first compliance-based design has a compliance of c = 1.95, and a final compliance of c = 1.46. The final stress-based mesh has 2878 elements, while the final compliance-based mesh has 2740 elements.



(a) Compliance design with mechanical and thermal loads

(b) Stress-based design with mechanical and thermal loads

Figure 3.6: Comparison between designs with only mechanical loads and combined mechanical and thermal loads

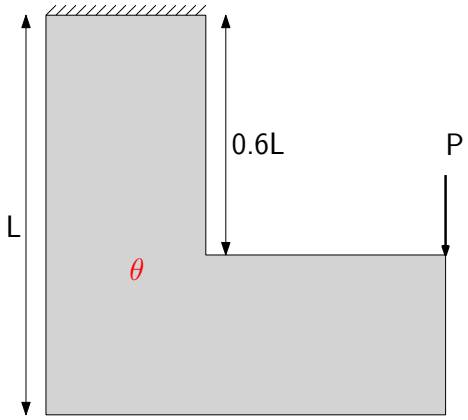


Figure 3.7: L-bracket domain subjected to traction loading

thermal loading.

The L-bracket problem shown in Figure 3.7 is a common topology optimization benchmark problem. Previous thermoelastic topology optimization results were shown in [75]. Similar to before, both compliance minimization and stress minimization problems are investigated. The L-bracket domain is clamped at one end and loaded at the other, with a total load denoted by P, which is modeled as a traction. The load P is applied as a 30 MPa traction along 5 mm of the end of the upper edge. The value of L is 0.2 m and the uniform change in temperature field θ is 10°C. The domain is 1 mm thick and the material chosen for this problem is 7075-T6 Aluminum and its properties are listed in Table 3.2. The domain is discretized in an unstructured manner, creating an initial mesh with 6736 elements. The design optimization

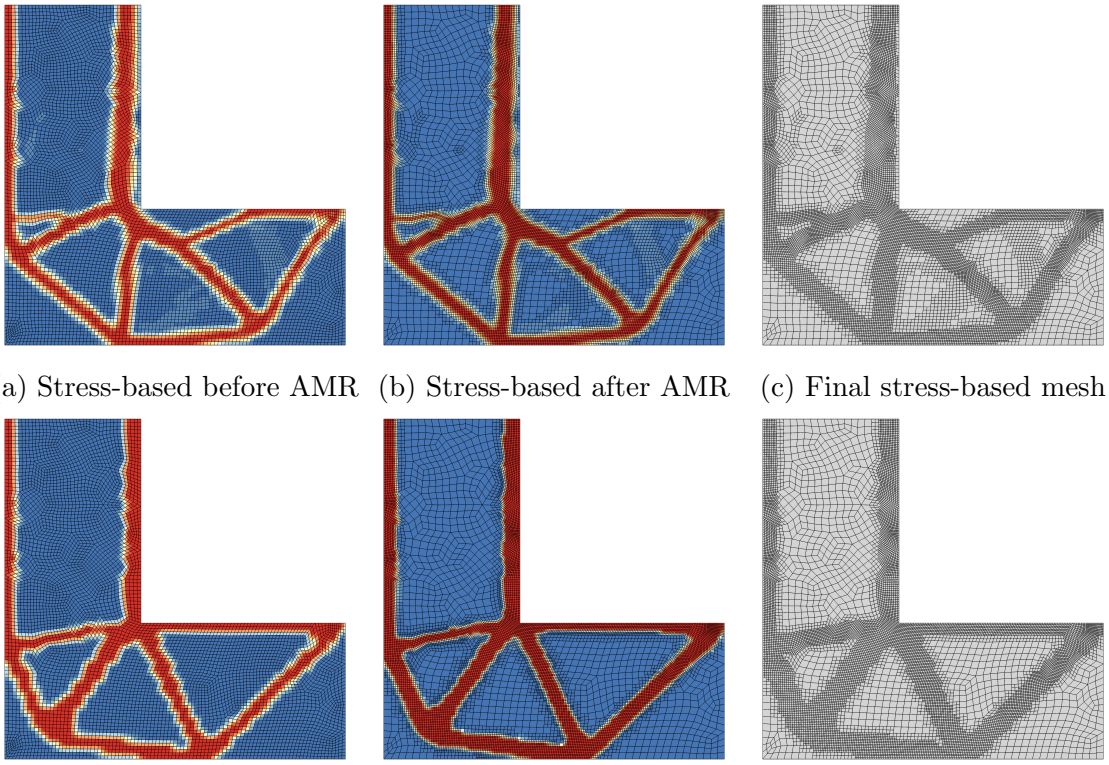
Table 3.2: Material properties of 7075-T6 Aluminum

Material Property	Value
E (GPa)	70.0
Poisson's ratio ν	0.33
Yield strength (MPa)	275.0
Coefficient of Thermal Expansion α (10^{-6}/°C)	23.5
Thermal Conductivity (W/(m°C))	130.0

is started from a uniform point with $\mathbf{x} = 0.95$. Adaptive mesh refinement is used once after the design converged on the initial mesh or after 500 optimization iterations. For the compliance-minimization problem, the volume constraint is restricted to 28.7%, which is the volume obtained from stress-constrained mass-minimization problem. Figure 3.8 shows the resulting design with solid denoted in red and void in blue.

Figures 3.8f and 3.8c show the mesh after one cycle of adaptive mesh refinement, which increases the number of elements to 11 839 and 12 328, respectively. Similar to the previous example, adaptive mesh refinement makes the boundary smoother and improves upon the objective function of the design at reduced computational cost. From Figures 3.8e and 3.8b, for the compliance-based design, the optimizer favored towards placing material along the reentry corner to maximize the stiffness of the structural layout whereas for the stress-based design, the optimizer avoided the reentry corner to avoid stress concentration, a trend that is well studied [133].

The third 2D example investigated is the form of the cantilever beam subjected to a point load as shown in Figure 3.9. For this problem, the plate is taken to be 1 cm thick, clamped at one end and loaded with point load P set at 5 kN. L is given to be 0.1 m and the uniform change in temperature field θ is set at 10°C. The material chosen for this problem is 4340 steel and its properties are listed in Table 3.1. The domain is discretized with an initial mesh size of 96 by 32 elements. A uniform initial design variable $\mathbf{x} = 0.95$ is used, and adaptive mesh refinement is employed once after



(a) Stress-based before AMR (b) Stress-based after AMR (c) Final stress-based mesh

(d) Compliance before AMR (e) Compliance after AMR (f) Final compliance mesh

Figure 3.8: Comparison of compliance-based and stress-based L-bracket designs with adaptive mesh refinement. The first stress based result has a volume of m = 0.292, and the final stress-based design has a volume of m = 0.287. The first compliance-based design has a compliance of c = 0.646, and a final compliance of c = 0.548. The final stress-based mesh has 12 328 elements, while the final compliance-based mesh has 11 839 elements.

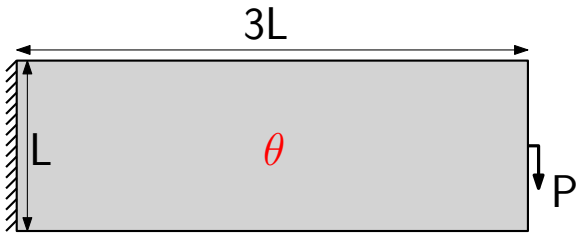
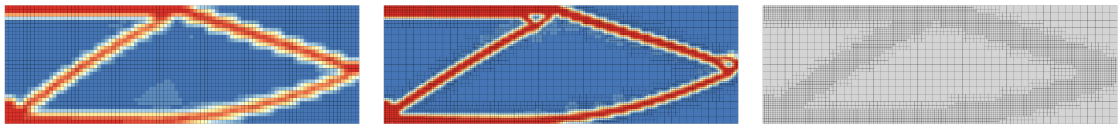
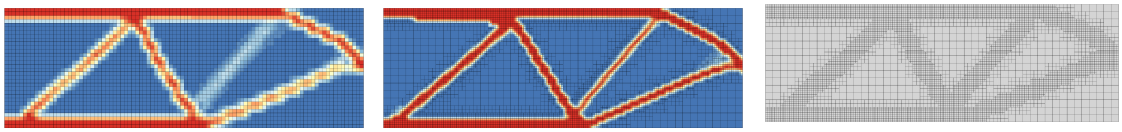


Figure 3.9: Cantilever beam with load P and uniform change in temperature field



(a) Stress-based before AMR (b) Stress-based after AMR (c) Final stress-based mesh



(d) Compliance before AMR (e) Compliance after AMR (f) Final compliance mesh

Figure 3.10: Comparison of compliance-based and stress-based cantilever beam designs with adaptive mesh refinement. The first stress based result has a volume of m = 0.262, and the final stress-based design has a volume of m = 0.237. The first compliance-based design has a compliance of c = 10.514, and a final compliance of c = 7.897. The final stress-based mesh has 5946 elements, while the final compliance-based mesh has 8169 elements.

the design converged on the initial mesh, or after a maximum of 250 optimization iterations. For the compliance-minimization problem, the volume constraint is set at 23.6%, which is the volume obtained from stress-constrained mass-minimization problem.

From Figures 3.10e, the compliance-based design with combined mechanical and thermal load is nonsymmetric whereas a compliance-based design with only mechanical load is symmetric. However, the design with thermal loads are created with more members under compression while the design with only mechanical loads has identical number of members under compression and tension. Another point to note, on the initial uniform mesh, both the compliance-based and stress-based designs produced design that are not fully discrete. With adaptive mesh refinement, these issues

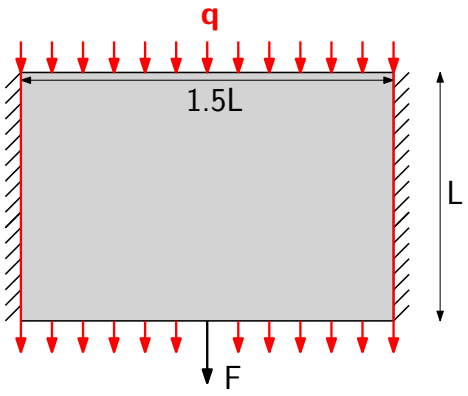


Figure 3.11: Bi-clamped domain with load P and nonuniform change in temperature field

are suppressed. Compliance-based design favored formation of slender members that are under compression while stress-based designs preferred the formation of holes to alleviate stress concentrations on an otherwise sharp corner.

This last 2D example showcase the capability of the current analysis where heat transfer component comes into play. The domain investigated, as shown in Figure 3.11, is similar to that bi-clamped plate domain, albeit with some changes. For this problem, only compliance-minimization, mass-constrained is carried out for the purpose of demonstration. For this problem, the plate is taken to be 1 cm thick, clamped at both ends and L is 0.04 m. The load F^m is modeled as a traction with magnitude 300 MPa, along a 5 mm edge which has its two endpoints equidistant from the midpoint of the lower edge. This is so that stress concentration is minimized. The nonuniform temperature change is due to the heat flux Q^{th} acting on the top and bottom edges as well as the two ends that are insulated such that $\theta = 0$. The magnitude of Q^{th} is 3000 Wm^{-1}. The material chosen for this problem is 4340 steel and its properties are listed in Table 3.1.

The domain is discretized into an initial mesh size of 88 by 64 elements, a total of 5632 elements and setting an uniform initial design variable of $\mathbf{x} = 0.95$. Adaptive mesh refinement is carried out once after the design converged on the initial mesh. A series of compliance-minimization designs with different mass constraints and their

temperature distributions are shown in Figure 3.12.

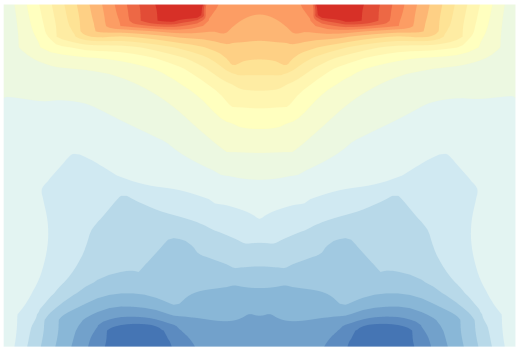
For different volume fraction constraint, the compliance-based designs produced are quite varied as it retains few identifiable features across volume fraction constraints. With this demonstration, other types of topology optimization objective can be explored such as the minimization of thermal compliance.

2D Thermoelastic Multimaterial Topology Optimization Results

In this section, results for multimaterial thermoelastic topology optimization are discussed. Design problems used in Section 3.4.1 are also used for multimaterial design. Similar to before, both compliance-based and stress-based designs are obtained using adaptive mesh refinement to compare against previous results [25]. Identical initial mesh sizes are used for both problems. To illustrate the difference in the structural layout for the two problems for the same mass, results from the stress-constrained mass-minimization problem are used as the corresponding mass for the mass-constrained compliance-minimization problem. Penalization parameters are set to values of $q_D = 8.0$, $q_\lambda = 0.0$, and $q_\beta = 8.0$, respectively. Two candidate materials, which were used in Gao, Xu, and Zhang [25], are used for results presented in this section and their properties are listed in Table 3.3. The two candidate materials are TC4, a low cost titanium alloy, and 18Mn2CrMoBA, a high-strength aviation steel. For simplicity of discussion, they are labeled M1 and M2 respectively. As detailed in [25], due to the presence of significant thermal load, the final optimal design will include both materials. While M2 is stiffer than M1 due to its larger Young's Modulus, it also has a larger value of GTSC, given by $E\alpha/1-2\nu$, indicating that M2 undergoes higher thermal loading for the same θ compared to M1. Therefore, depending on the influence of the thermal loading, the proportion of material in the final design can change.

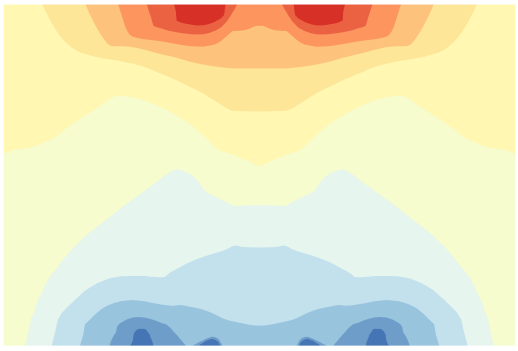
The first problem examined here is the bi-clamped plate problem shown in Fig-

(a) Compliance-based with volume fraction of 35%



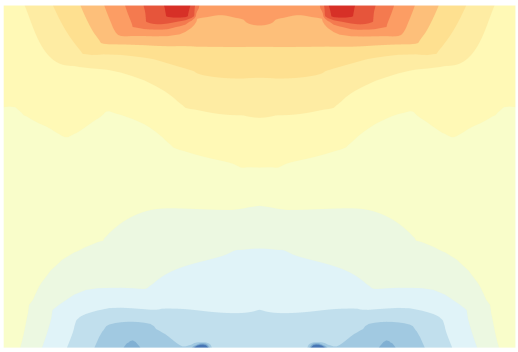
(b) Temperature distribution with $\theta_{\max} = 6.6°\text{C}$ and $\theta_{\min} = -4.0°\text{C}$

(c) Compliance-based with volume fraction of 45%



(d) Temperature distribution with $\theta_{\max} = 4.9°\text{C}$ and $\theta_{\min} = -6.0°\text{C}$

(e) Compliance-based with volume fraction of 55%



(f) Temperature distribution with $\theta_{\max} = 3.7°\text{C}$ and $\theta_{\min} = -4.0°\text{C}$

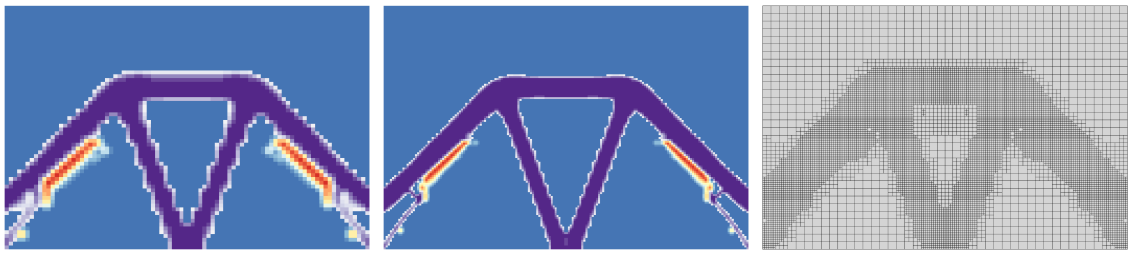
Figure 3.12: Comparison of compliance-based designs with varying volume fraction using adaptive mesh refinement.

Table 3.3: Candidate material properties used in 2D multimaterial topology optimization

Material Property	M1	M2
E (GPa)	105	190
Density (kg/m^3)	4440	7850
Poisson's ratio ν	0.34	0.28
Yield strength (MPa)	880	350
Coefficient of Thermal Expansion α (10^{-6}/°C)	9.1	12.4
Thermal Conductivity (W/(m°C))	8.7	35.38
GTSC β (MPa/°C)	2.986	5.355

ure 3.4. Multimaterial results were shown by Gao, Xu, and Zhang [25]. For this problem, the plate is taken to be 1 cm thick, clamped at both ends and L is 0.04 m. The load P is modeled as a traction with magnitude 260 MPa, along a 5 mm edge which has its two endpoints equidistant from the midpoint of the lower edge. This loading alleviates a stress concentration that would arise if a point-load were applied. The uniform change in temperature field θ is 25°C. The domain is discretized with an initial mesh size of 88×64 elements with a total mesh size of 5632 elements. The two material design variables for all elements are set at $\mathbf{x}_i = 0.45$, topology design variable $t = 0.9$. One cycle of adaptive mesh refinement is applied after the design converges on the initial mesh. An element is refined if the topology variable is less than the value of $x_{ru} = 0.9$ and is coarsened if the topology variable is greater than $x_{rl} = 0.95$. The mass-constrained, compliance-minimization problem is subjected to a fixed normalized mass constraint of $m_{\text{fixed}} = 0.2624$, which is the ratio of mass obtained from mass-minimization, stress-constrained problem to the product of the average density of M1 and M2 with the volume of the domain. Figure 3.13 shows the resulting design with M1 denoted in red, M2 denoted in purple, and void in blue.

Figure 3.13, shows that the compliance-based and stress-based designs exhibit markedly different topologies, even for the same mass. Unlike in the case of single material stress-constrained mass-minimization, which is essentially volume-minimization,



(a) Stress-based before AMR (b) Stress-based after AMR (c) Final stress-based mesh

(d) Compliance before AMR (e) Compliance after AMR (f) Final compliance mesh

Figure 3.13: Comparison between the stress-based and compliance-based bi-clamped plate multimaterial designs with adaptive mesh refinement. The first stress based result has a mass of m = 0.3347, and the final stress-based design has a normalized mass of m = 0.2624. The first compliance-based design has a compliance of c = 1.134, and a final compliance of c = 1.013. The final stress-based mesh has 10 246 elements, while the final compliance-based mesh has 10 372 elements.

multi-material stress-constrained mass-minimization provides more design freedom to the optimizer. For compliance-based design compression is favored to act against the thermal expansion. The optimizer therefore places more of the structure under compression, and the results follow those shown by by Gao, Xu, and Zhang [25]. On the other hand, members under compression create thermal stresses, thus they are disfavored and are slender in the final stress-based design. Also, two truss-like tension members were formed in the design to alleviate thermal stress. With feature-based adaptive mesh refinement, the adaptively refined mesh of Figures 3.13e and 3.13b are centered around the structural layout where additional elements are added to the existing material feature and removed from voids. The resulting mesh produced a design with higher resolution features, especially in stress-constrained design as shown by the differences in Figures 3.13b and 3.13c. The objective values of the compliance-based and stress-based designs also improved with adaptive mesh refinement, with a

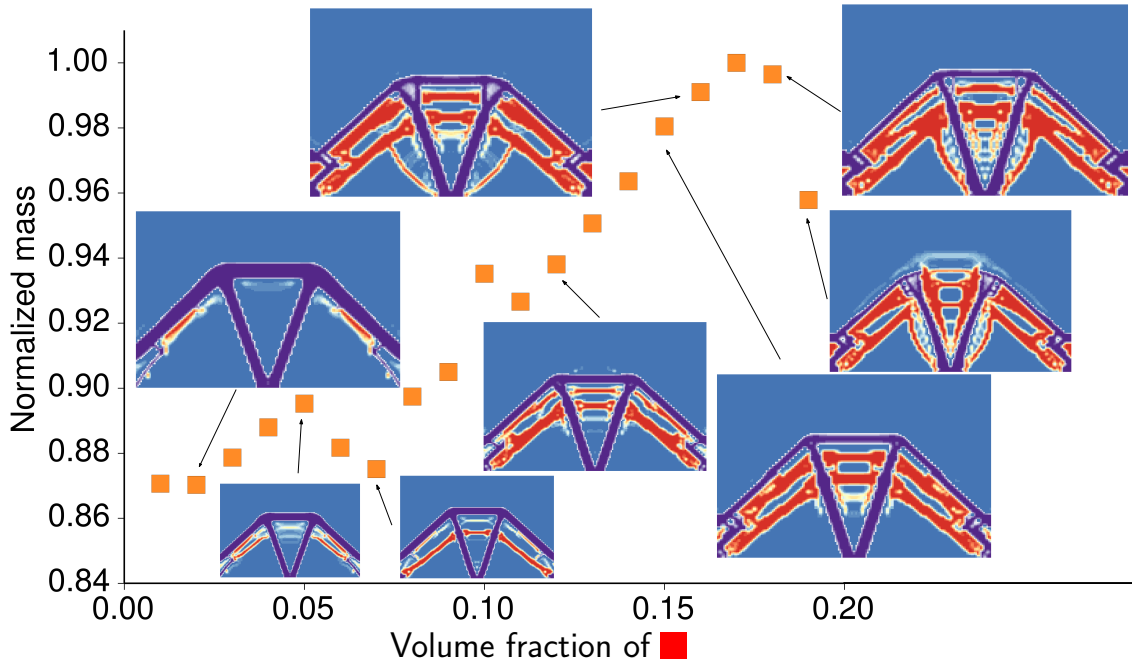


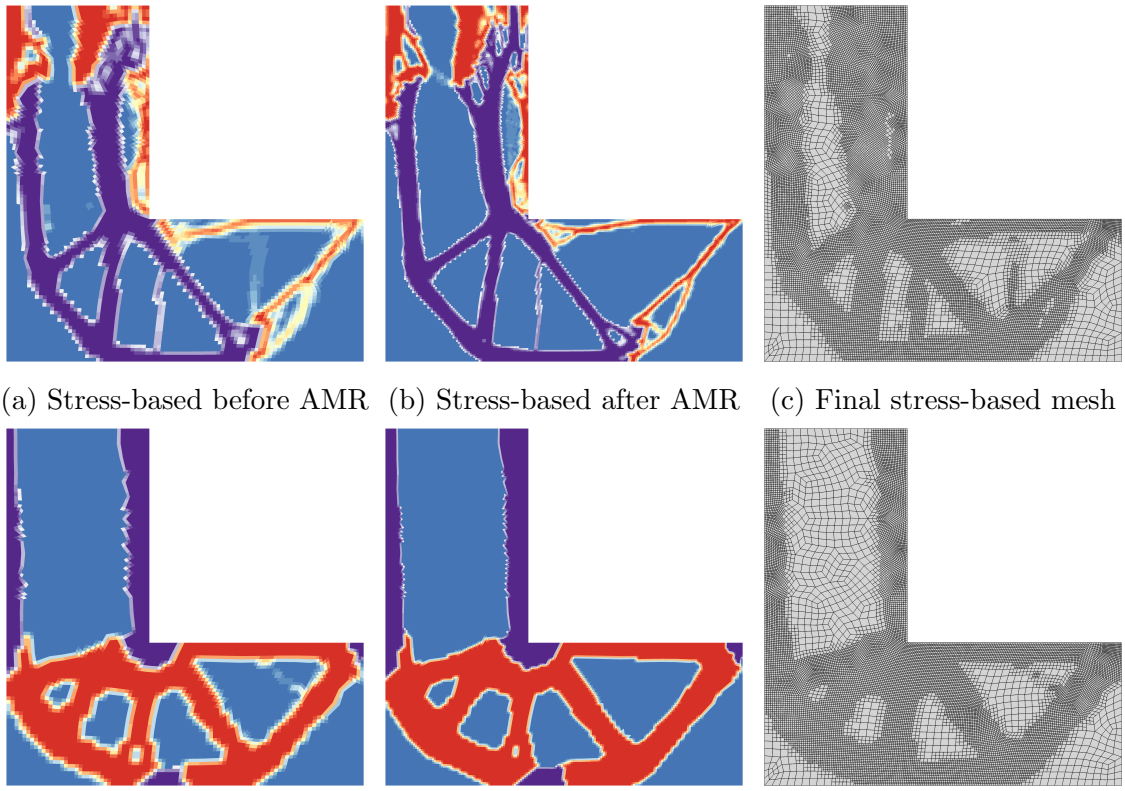
Figure 3.14: Effects on the final stress-based design with varying minimum volume requirement on M1.

reduction of 10.7% in compliance and 19.6% in mass respectively.

Figure 3.13 shows that the presence of M1 is minimal in the stress-based design. In order to have a better understanding of the presence of M1 on the final design as well as investigating the robustness of the framework, a series of mass-minimization, stress-constrained optimization problems with different minimum volume fraction on M1 are carried out. The resulting designs and its corresponding mass is shown in Figure 3.14. Figure 3.14 shown that, with some exceptions, a general trend of increasing mass as the minimum volume requirement on M1 increased as expected. There are two local minima designs, one of which is the design shown in Figure 3.13, which has no minimum volume fraction requirement on M1 as well as the design with a 5% minimum volume requirement. Beyond the 5% design, the mass of the 6%, 7%, 18% and 19% designs dropped significantly, which can be attributed to the M1 structure in the final design not fully converging, giving us a lower mass. As the minimum volume requirement on M1 increased from 8% to 15%, the variation in the design

remains consistent as the optimizer added to more material to the existing structure. For the designs with 16% to 19% minimum volume requirement on M1, additional M1 trusses are formed as well as a significant reduction in the volume of M2 material in the final design.

The second design problem is the L-bracket problem shown in Figure 3.7. Similar to before, both compliance minimization as well as stress-constrained mass-minimization problems are examined. The L-bracket is clamped at the top left and loaded at the upper right, denoted by P, which is modeled as a traction. The load P is applied as a 62.5 MPa traction along 5 mm of the end of the edge, the value of L is 0.1 m, and the uniform change in temperature field θ is 10°C. The domain is modeled to be of unit thickness and the candidate materials are shown in Table 3.3. The domain is discretized into an initial mesh size of 6736 elements, with the two material design variables for all elements are set at $\mathbf{x}_i = 0.45$, topology design variable $t = 0.9$ and adaptive mesh refinement is carried out once after the design converged on the initial mesh. The mass-constrained compliance-minimization problem is subjected to a fixed normalized mass constraint of $m_{\text{fixed}} = 0.4043$, which is the ratio of mass obtained from stress-constrained mass-minimization problem to the product of the average densities of M1 and M2 with the volume of the domain. Adaptive mesh refinement is carried out once on the initial mesh using refinement criterion of $x_{ru} = 0.9$ and $x_{rl} = 0.95$. Figure 3.15 shows the resulting design with M1 denoted in red, M2 denoted in purple, and void in blue. Similar to the previous example, Figure 3.15 shows that compliance-based and stress-based designs produced vastly different topologies for the same mass. For compliance-based design, the optimizer preferred the stiffer M2 along the re-entry corner and the edge of the traction application to sustain the bending moment for improved compliance. On the other hand, for the stress-based design, the optimizer seeks to avoid the re-entry corner to minimize stress concentration. Also, the optimizer placed more of M1, which has the higher



(a) Stress-based before AMR (b) Stress-based after AMR (c) Final stress-based mesh

(d) Compliance before AMR (e) Compliance after AMR (f) Final compliance mesh

Figure 3.15: Comparison of stress-based and compliance-based L-bracket multimaterial designs with adaptive mesh refinement. The first stress based result has a mass of m = 0.5162, and the final stress-based design has a mass of m = 0.4043. The first compliance-based design has a compliance of c = 0.1729, and a final compliance of c = 0.1517. The final stress-based mesh has 18 193 elements, while the final compliance-based mesh has 15 685 elements.

Table 3.4: Candidate isotropic material properties.

Material Property	M1	M2
Young's Modulus E (GPa)	70	35
Density ρ (kg/m^3)	2600	1300
Poisson's ratio ν	0.3	0.3

Table 3.5: Orthotropic material properties.

Material Properties	M3
Young's Modulus E_{xx} (GPa)	70
Young's Modulus E_{yy}, E_{zz} (GPa)	23.3
Shear Modulus G_{yz} (GPa)	10
Density ρ (kg/m^3)	2600
Poisson's ratio ν_{xy}, ν_{xz}, ν_{yz}	0.3

yield stress, along the corner as a form of buffer to M2. As with before, the adaptively refined mesh produced a design with smoother structural boundary and higher resolution features in the stress-based design. The objective values of the compliance-based and stress-based designs also improved significantly using adaptive mesh refinement, with a reduction of 12.3% in compliance and 21.7% in mass respectively.

3D Multimaterial Topology Optimization Results

Three different 3D design problems are investigated to demonstrate the capability of the framework for large-scale topology optimization: a cantilever, an orthogonal bracket, and an orthotropic box. The first two problems investigated are isotropic material selection problems where there are two candidate materials, while the third problem involves the selection of the principal direction of an orthotropic material with 13 candidate orientations. For the isotropic cases, the two candidate material properties are listed in Table 3.4, while the material properties for the orthotropic material are listed in Table 3.5.

The first isotropic example is a cantilever beam of overall length L shown in Figure 3.16. A hole with diameter 0.1L is cut out at 0.85L from the root of the beam.

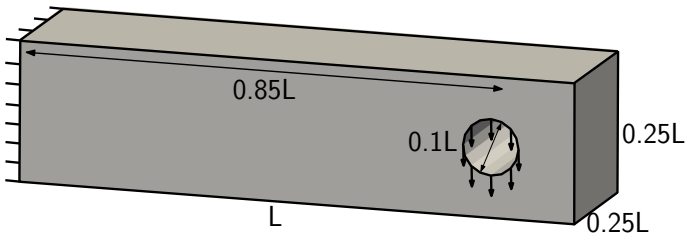
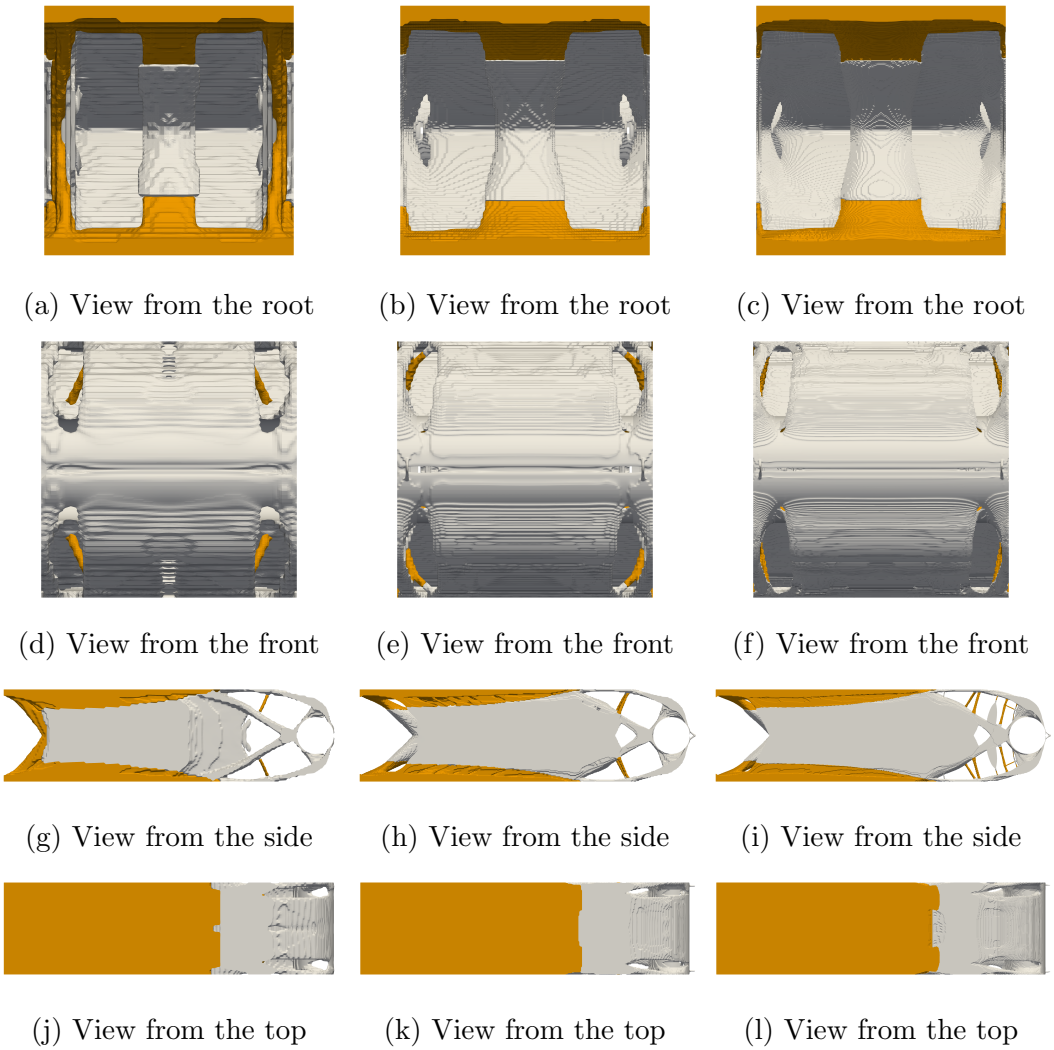


Figure 3.16: Dimensions for the cantilever beam problem.

A traction is applied in the vertical direction throughout the hole. In this work, the value of L is 1 m and the magnitude of the applied traction is 5.8 MPa. The mass is fixed at $0.2\rho_{\text{average}}V$, where V is the volume of the domain and ρ_{average} is the average density of the candidate materials used.

For this problem, an external planar mesh, generated in NX [136], is imported into TMR which is used to generate the hexahedral mesh. Three different meshes are used for this study: a medium adaptive mesh, a fine adaptive mesh, and an ultra fine fixed mesh case. The initial mesh for the medium adaptive case has 5.14 million hexahedral elements, 5.27 million nodes, and 2.02 million design variables. After the adaptive refinement step, the final mesh contains 11.3 million elements, 11.0 million nodes, and 4.59 million design variables. The initial mesh for the second adaptive case has 41.1 million elements, 41.6 million nodes, and 15.8 million design variables. After the adaptive refinement step, the final mesh contains 81.8 million elements, 80.3 million nodes and 31.8 million design variables. The fixed mesh without adaptive refinement has 329 million elements, 331 million nodes, and 125 million design variables. For the ultra fine problem, multigrid-preconditioned GMRES(50) with a maximum of five restarts is employed to solve the governing equations [109] to save memory. The final design for each case is shown in Figure 3.17. Figure 3.17 shows the resulting designs for the three cases with M1 denoted in orange and M2 denoted in gray. For all three cases, the optimizer produced a multi-section beam design with a large central web and two thinner outer webs arranged symmetrically about the center line of the beam.



(a) View from the root (b) View from the root (c) View from the root

(d) View from the front (e) View from the front (f) View from the front

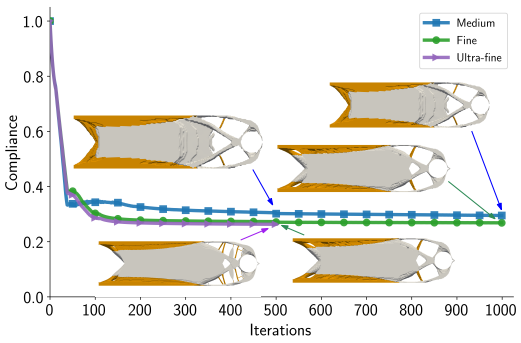
(g) View from the side (h) View from the side (i) View from the side

(j) View from the top (k) View from the top (l) View from the top

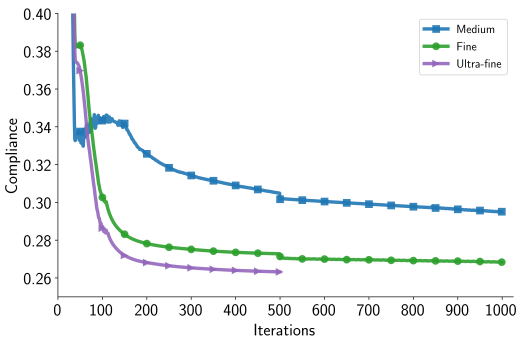
Figure 3.17: Final design of the beam from the three cases with M1 denoted in orange and M2 denoted in gray. From left, center to right are the medium resolution case, the fine resolution, and the ultra-fine resolution.

The stiffer M1 material extends from the cantilever root along the top and bottom of the structure, increasing its bending stiffness. The softer and lighter M2 material forms the shear webs of the beam, connecting the stiffer M1 material at the top and bottom of the structure and also forms the structure surrounding the load application point. There is a slight difference in the distribution of M1 at the cantilever root for the medium case compared to the fine and ultra-fine case. As the resolution of the initial mesh becomes finer, the design at the front of the structure towards the load application point becomes more intricate and slender. The meshes for the medium and fine design cases do not have the resolution to form the structure as shown in the ultra-fine case.

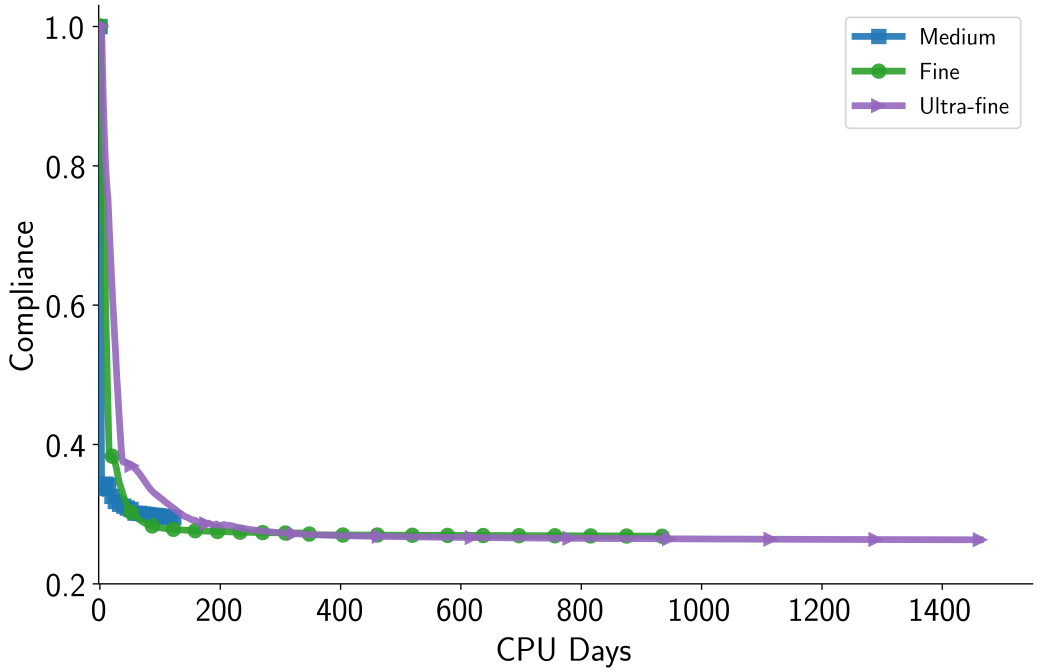
Figure 3.18 shows the convergence of the three cases as a function of iteration history. The compliance value shown is normalized by the initial compliance of the medium resolution design. For an accurate comparison, the final adaptive mesh refinement design is projected onto the uniformly fine mesh for compliance evaluation and found that the effect of the mesh, especially at this resolution, on the compliance is in the order of 0.1%. For the medium resolution case, the problem is solved using 96 processors, requiring 1000 optimization iterations over 30.7 hours. For the fine resolution case, the problem is solved using 456 processors, requiring 1000 optimization iterations over 49.2 hours. For the ultra-fine case, the problem is solved using 456 processors, requiring 500 optimization iterations over 77.0 hours. To compare the computational effort for each case, the CPU days, which is the product of the number of processors with the optimization wall time, is used as the benchmark. The convergence history of the compliance of these three cases are plotted as a function of its CPU days as shown in Figure 3.18c. A summary of the optimization outcome for the three cases are in Table 3.6. From Figure 3.18a, I observed that there are some slight differences in the normalized compliance of the three different designs. These differences are illustrated in a close-up as shown in Figure 3.18b and summarized



(a) Compliance convergence history for the beam cases.



(b) Zoom view for the compliance convergence history for the beam cases.



(c) Compliance convergence for the beam cases as a function of CPU days

Figure 3.18: Convergence history of the beam design for the three resolution cases.

Table 3.6: Summary of optimization outcomes for each design case.

Optimization Outcome	Medium	Fine	Ultra-fine
Normalized Compliance	0.295	0.268	0.263
CPU Cores	96	456	456
Walltime (h)	30.7	49.2	77.0
Elements in initial mesh (million)	5.14	41.1	329
Elements in adaptively refined mesh (million)	11.3	81.8	-
Relative resolution of final design (10^{-3})	3.06	1.53	1.53
CPU Days	122.7	934.4	1463.2

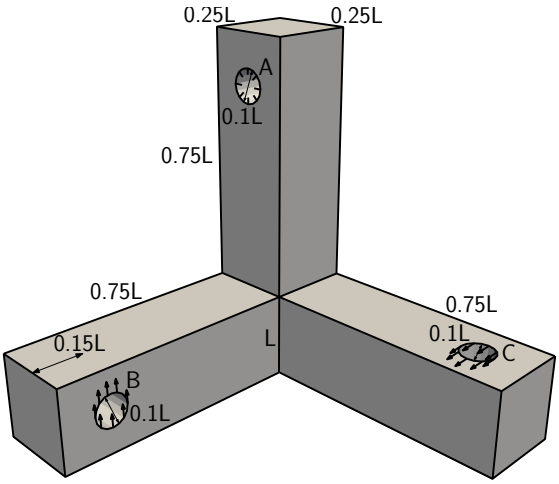
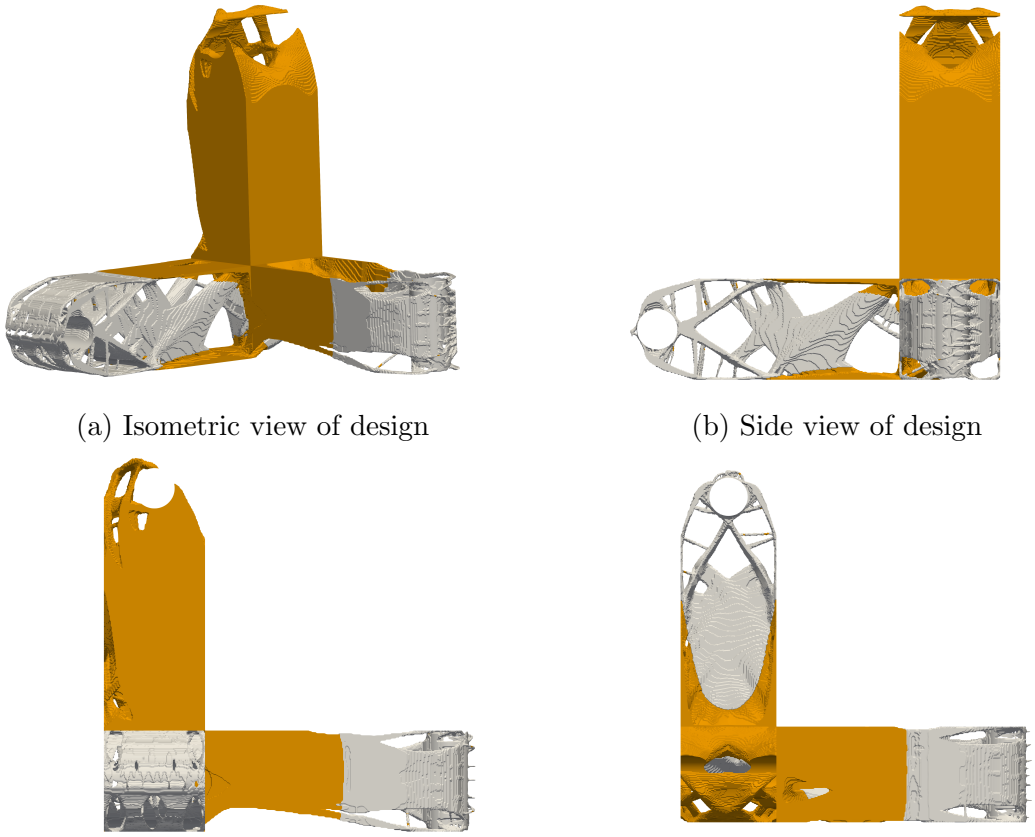


Figure 3.19: Orthogonal bracket problem domain

in Table 3.6. The compliance of the medium resolution case is 10.8% larger when compared to the ultra-fine resolution case while the compliance of the fine resolution design is 1.86% larger when compared to the ultra-fine resolution case. This is not surprising when accounting for the larger difference in distribution of the materials in the medium resolution design with the ultra-fine resolution design as compared to the fine and ultra-fine resolution designs, given that the medium resolution design has fewer design freedom for the optimizer. Figure 3.18c illustrated the advantage of the adaptive mesh refinement approach over the fixed mesh. Note that to obtain the ultra-fine resolution design, it took 1463.2 CPU days whereas to obtain the fine resolution design, which has identical relative resolution compared to the ultra-fine resolution design, took only 934.4 days, signifying an improvement of 36.1% in computational resources used and only 1.86% different in terms of compliance. Lastly, the medium resolution design took only 122.7 CPU days while having a final relative resolution twice that of the fine and ultra-fine resolution designs, a difference of 91.6% in terms of computational resources used. For all three designs, the mass constraint is satisfied to machine precision.

The second problem investigated is the orthogonal bracket. The geometry of the orthogonal bracket is shown in Figure 3.19. It consists of 3 identical orthogonal beam



(a) Isometric view of design

(b) Side view of design

(c) Front view of design

(d) Top view of design

Figure 3.20: Final design of the orthogonal bracket with M1 denoted in orange and M2 denoted in gray.

members of length L, each with a cylindrical holes cut 0.15L from the free ends of each member. Hole A at the top of the vertical member is completely clamped, and traction loads of magnitude 3.2 MPa are applied to holes B and C. The loads on B and C create a bending moment in each of the horizontal members, and a combined bending and torsional load in the vertical member. Here, the length L shown in the diagram below is 1.0 m. The mass constraint of the compliance-minimization problem is fixed at $0.2\rho_{\text{average}}V$, where V is the volume of the domain and ρ_{average} is the average density of the candidate materials used.

The mesh used in this study has 77.0 million elements, 77.9 million nodes, and 29.5 million design variables. No adaptive mesh refinement is used for this problem. The

optimization took 58 hours on 192 processors over 500 iterations. For this problem, 5 multigrid levels are used. Figure 3.20 shows the resulting design with M1 denoted in orange and M2 denoted in gray. Note that the stiffer material M1 is placed primarily along the vertical member and the re-entrant corner to counteract the combined bending and torsional loads. Along the horizontal members where the primary loading is due to bending, M2, which is lighter and less stiff, is used to minimize the deflection as well as satisfy the mass constraint. Figure 3.21 shows the convergence of the optimization and the overall design history with the compliance value normalized by its initial value. The normalized compliance rapidly decreases over the first 50 iterations, while the mass constraint violation initially exceeds 50%. From iteration 100 to 330, significant changes in the design occurred mainly along the horizontal members, with more truss-like members forming as the optimizer seeks to reduce the mass of the design to satisfy the constraint.

Figure 3.22 shows detailed designs at design iterations: 150, 300, and 400, respectively. Figure 3.22a and 3.22b, show where more truss members started to form. From iteration 330 to 500, with the constraint satisfied, the optimizer created designs with intricate trusses, as shown in Figure 3.22c. Note that without this mesh resolution, the optimizer would not have been able to form them.

For the last example, the design problem involves the placement and orientation of an orthotropic material within a rectangular domain. The multimaterial design in this case requires the optimizer to choose the alignment of the material along the different possible directions. Figure 3.23 shows the 13 possible directions for each element. The orthotropic properties of the material are listed in Table 3.5.

The rectangular design domain is shown in Figure 3.24. The bottom 4 corners of the domain are pinned and a traction load is applied on the bottom of the box over a small square section, with area $0.125\mathrm{L}^2$, centered on the bottom surface. The dimension L is 0.1 m and the traction load applied is 60 kPa. Since the problem is

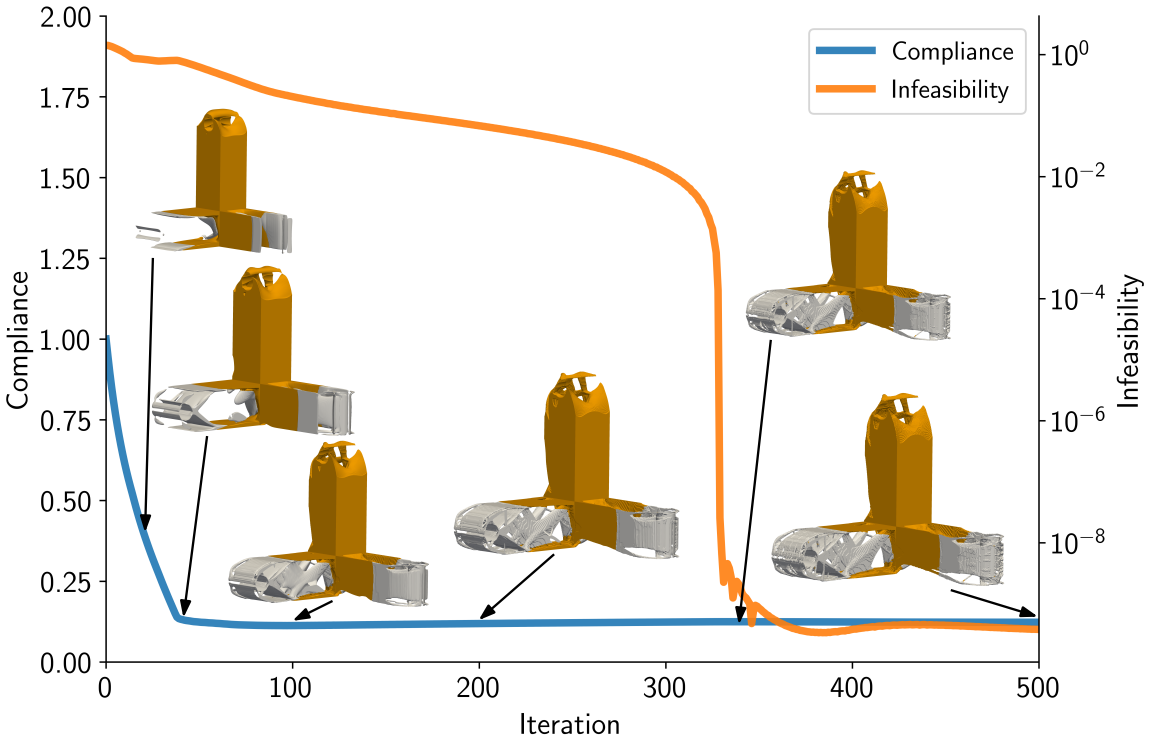
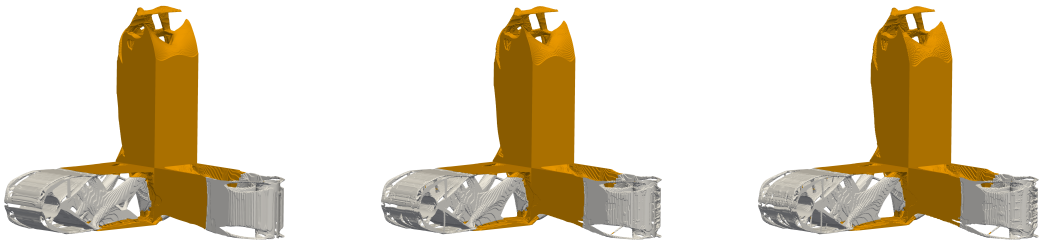


Figure 3.21: Orthogonal bracket convergence and design history

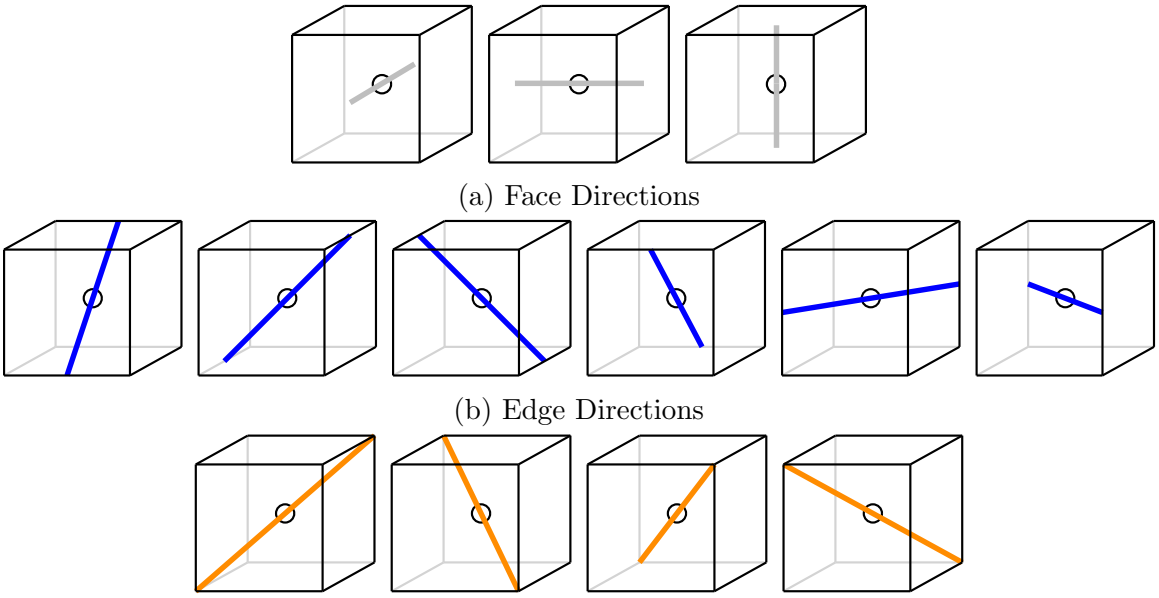


(a) Design at 150-th iteration

(b) Design at 300-th iteration

(c) Design at 400-th iteration

Figure 3.22: Designs of the orthogonal bracket at various optimization iterations



(a) Face Directions

(b) Edge Directions

(c) Corner Directions

Figure 3.23: Schematic of the 13 orthotropic directions

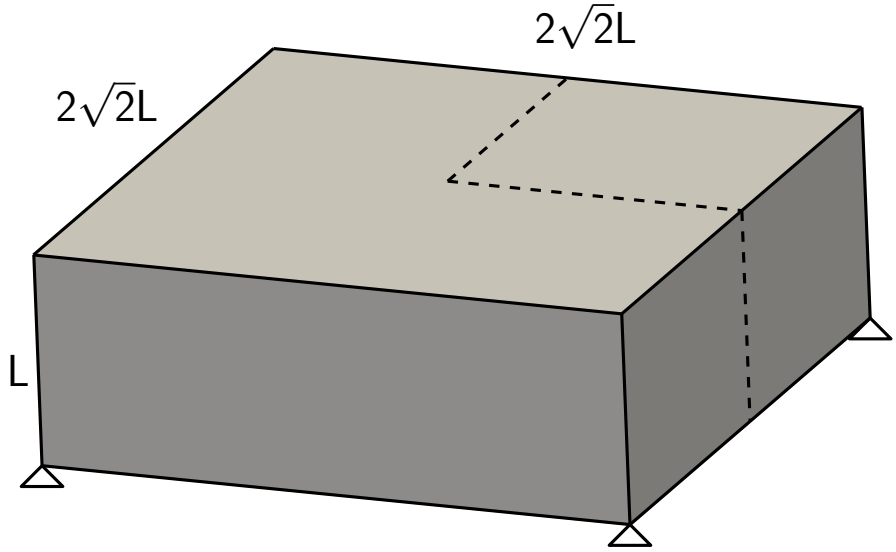
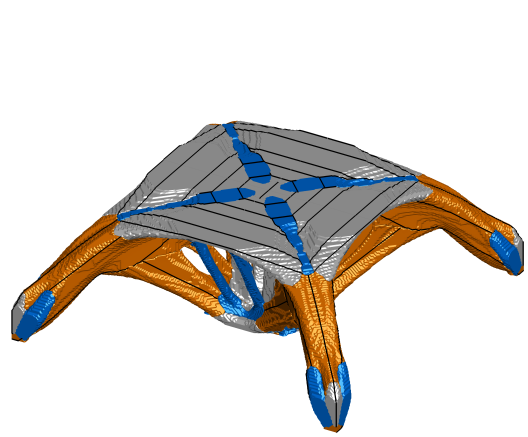


Figure 3.24: Schematic of orthotropic domain problem

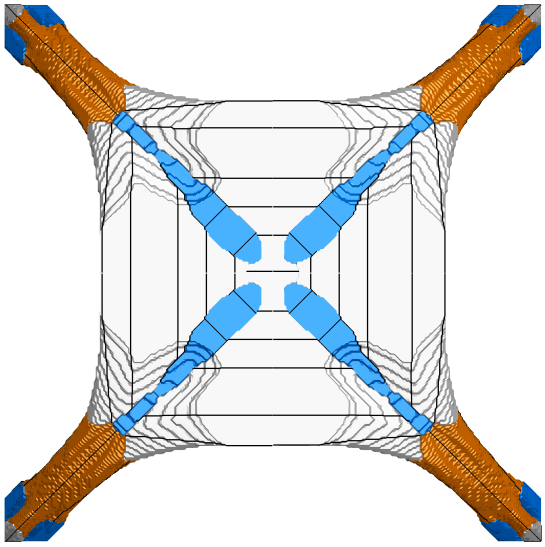
symmetric, symmetry is imposed and carry out design only a quarter of the domain. The initial mesh has 1.57 million hexahedral elements, 1.61 million nodes, and 2.90 million design variables. The mass constraint is fixed at 15% mass fraction. One cycle of adaptive mesh refinement is used and in this case and no mesh coarsening is applied. The problem is solved using 72 processors, requiring 1000 optimization iterations over 9.5 hours. For this problem, 4 multigrid levels are used. Figure 3.25 shows the final design, with the gray regions denoting the face directions, blue regions denoting the edge directions and orange regions denoting the corner directions. The lines on the structure denote the principal direction of the material.

After one cycle of adaptive mesh refinement, the final mesh had 3.19 million elements, 3.15 million nodes, and 5.91 million design variables. The final design utilized 12 out of the 13 orthotropic directions, leaving out only the edge direction in the positive x-y plane. Figure 3.26 shows the convergence history of the design as a function of the wall time, with every 50th iteration is indicated by a symbol. Figure 3.27 shows the design history as a function of and optimization iteration. The compliance value shown is the compliance normalized by the initial compliance.

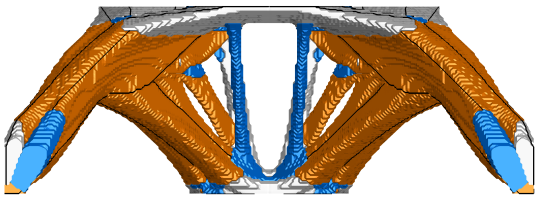
The total wall time for this optimization is 9.5 hours, with the adaptive mesh refinement step taking place at 4.1 hours into the optimization. Similar to the other design problems, the normalized compliance and the infeasibility of the design quickly converges. The design becomes feasible within the first 40 iterations and remains feasible for the rest of the optimization process. However, not much of the structure is formed at this design point as shown by Figure 3.27a. From iteration 60 onwards, the change in the normalized compliance becomes small and the structure starts to take shape as shown in Figure 3.27b and 3.27c. After 500 design iterations, the mesh is adaptively refined and the optimization is continued on this new mesh. The normalized compliance increases on the finer mesh and remains at a larger value, most likely due to the larger discretization in the orthotropic problem.



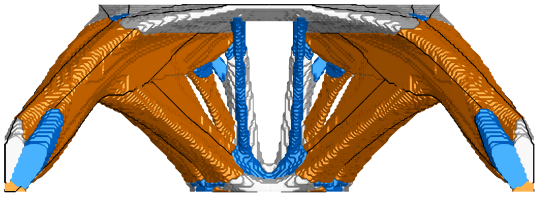
(a) Isometric view of design



(b) Top view of design



(c) View from the x-z plane



(d) View from the y-z plane

Figure 3.25: Final design of the orthotropic domain with M1 denoted in orange and M2 denoted in gray.

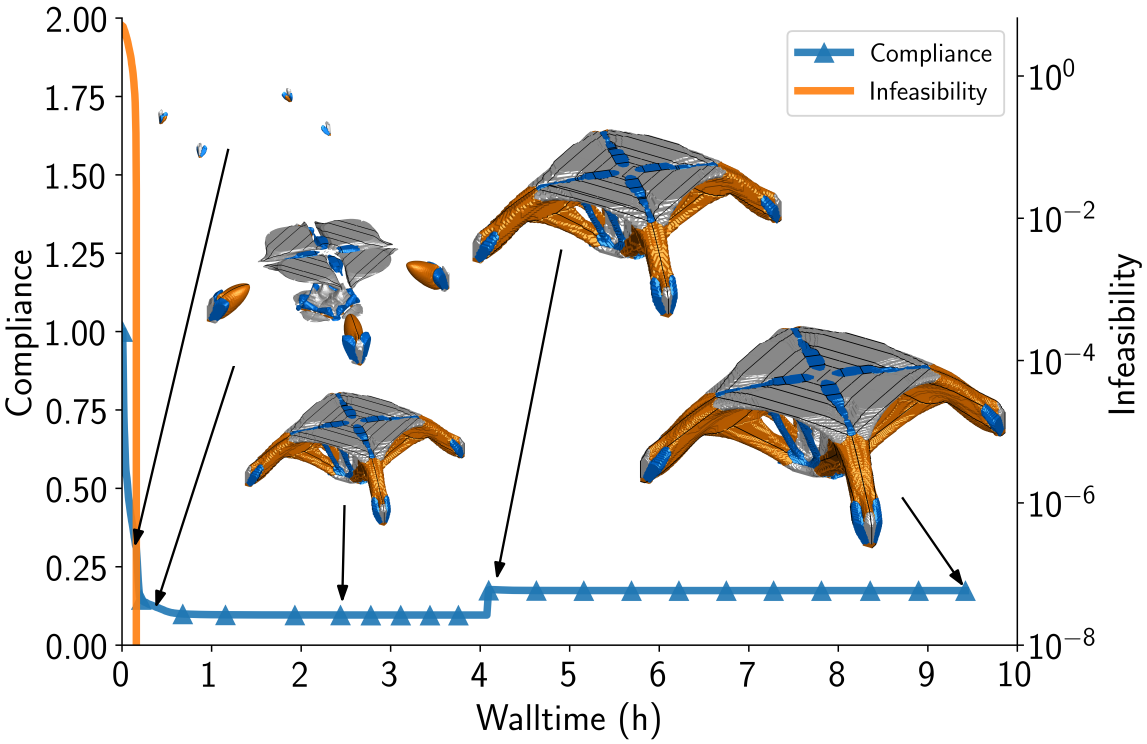
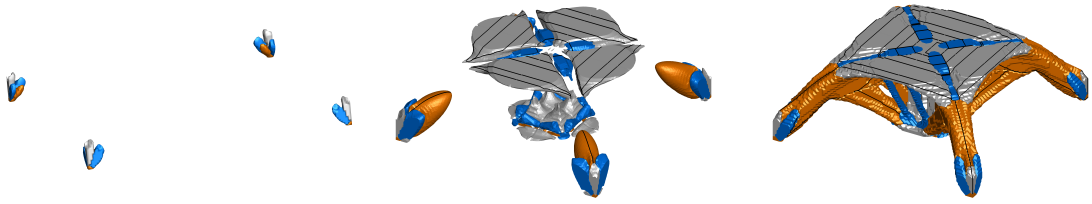


Figure 3.26: Orthotropic problem convergence history



(a) Design at 40-th iteration

(b) Design at 60-th iteration

(c) Design at 250-th iteration

Figure 3.27: Designs of the rectangular domain at various optimization iterations

Through this work, methods to obtain multimaterial topology optimization with and without thermal loads are shown. In particular, techniques employed to tackle large-scale multimaterial design problems are demonstrated. While large-scale topology optimization designs are desirable, they are difficult to create using uniform meshes due to significant computational cost. The framework shown here has the unique capability of generating large-scale quadrilateral and hexahedral meshes directly from CAD geometry and perform multimaterial topology optimization with mesh adaption. Both 2D and 3D single-material and multimaterial designs with adaptive mesh refinement were illustrated.

CHAPTER 4
HIGHER ORDER TOPOLOGY OPTIMIZATION USING B-SPLINE POLYNOMIALS

In this chapter, the development of B-spline polynomials and Bernstein polynomials for the use in higher order thermoelastic topology optimization framework is discussed. First, the work using Truncated Hierarchical B-splines (THB-splines) with topology optimization is illustrated. Then, the extension to higher order analysis and design within the framework through Bernstein polynomials is described. Lastly, the results to both methods are presented.

THB-spline with Topology Optimization

THB-splines are a subset of B-spline polynomials that are well-suited to adaptive mesh refinement and amenable to its application in topology optimization. THB-splines use a hierarchical basis that ensures compatibility across adjacent elements of different size. This hierarchical basis requires a set of local constraints between the weights on the B-spline basis functions. To ensure that the finite-element implementation is simple and efficient requires a careful consideration of the treatment of these compatibility constraints. For finite-element analysis using Lagrange polynomials, the number of active shape functions in each element and is constant for each element type of a given order. For THB-spline polynomials, the truncation across elements results in a varying number of active shape functions that span across grid levels. Some implementations of THB-splines require a series of different element implementations for boundary elements between coarse and fine elements. However, in the implementation presented here, the basis functions on each element are a straightforward B-spline basis that can be computed efficiently, while the truncation of the B-spline

functions are externally imposed in the form of constraints. These constraints are similar to hanging node constraints for adaptive mesh refinement associated with Lagrange basis functions. These constraints allow this approach to be compatible with existing FEM layout.

A p-th degree B-spline univariate curve is defined [137] as follows:

$$u(\mathbf{t}) = \sum_{i=0}^{n} N_{i,p}(\mathbf{t})\mathbf{t}_i \tag{4.1}$$

$\mathbf{t}_i$ is the i-th control point and $N_{i,p}(\mathbf{t})$ is the i-th B-spline basis function defined on a non-periodic and non-uniform knot vector $t = \{t_0, t_1, ..., t_m\}$ with m+1 knots. In the numerical implementation, the basis functions can be evaluated recursively defined using the Cox-de Boor's recursion formula [138]

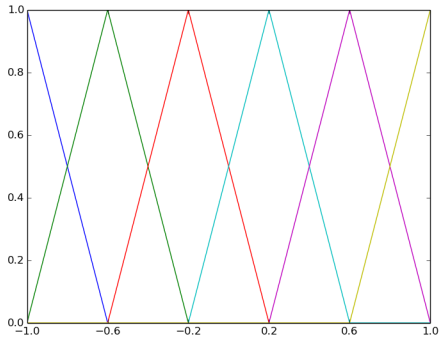
$$\begin{aligned} N_{i,p}(\mathbf{t}) &= \frac{(t - t_i)N_{i,p-1}(\mathbf{t})}{t_{i+p} - t_i} + \frac{(t_{i+p+1} - t)N_{i+1,p-1}(\mathbf{t})}{t_{i+p+1} - t_{i+1}} \\ N_{i,0}(\mathbf{t}) &= \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.2}$$

For FEM applications, the first and second derivative of the basis functions are of interest. The k-th derivative of a p-th degree B-spline polynomial on knot vector $t^{(k)} = \{t_0, t_1, ..., t_{m-2k}\}$ with $m + 1 - 2k$ knots can be recursively evaluated as:

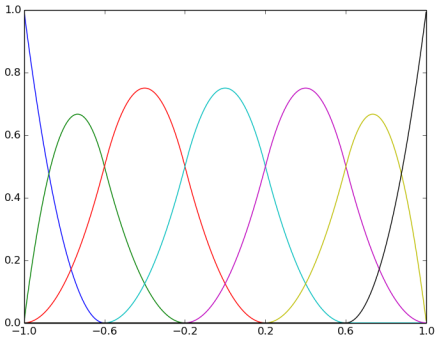
$$\begin{aligned} u^{(k)}(\mathbf{t}) &= \sum_{i=0}^{n-k} N_{i,p-k}(\mathbf{t}) t_i^{(k)} \\ t_i^{(k)} &= \begin{cases} t_i & k = 0 \\ \frac{p-k+1}{t_{i+p+1} - t_{i+k}} \left(t_{i+1}^{(k-1)} - t_i^{(k-1)} \right) & k > 0 \end{cases} \end{aligned} \tag{4.3}$$

The important properties of B-spline polynomials are summarized as follows [139]:

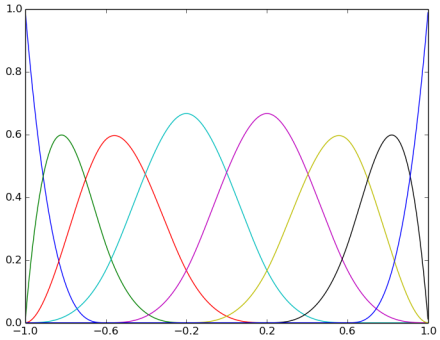
- Positivity and local support: The basis function $N_{i,p}$ is nonzero on at most



(a) Linear B-spline shape functions



(b) Quadratic B-spline shape functions



(c) Cubic B-spline shape functions

Figure 4.1: Univariate B-spline of varying polynomial degrees

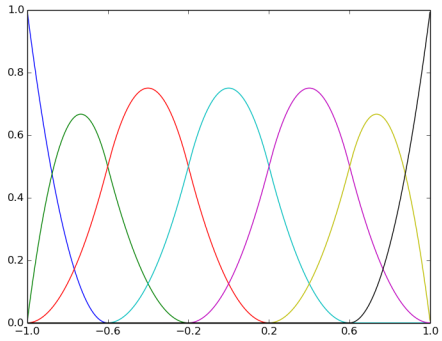
$(p+1)$ knot spans and vanishes outside of it

- Smoothness: $N_{i,p}$ is $(p-1)$ times differentiable with discontinuities of the p-th derivative at the control points

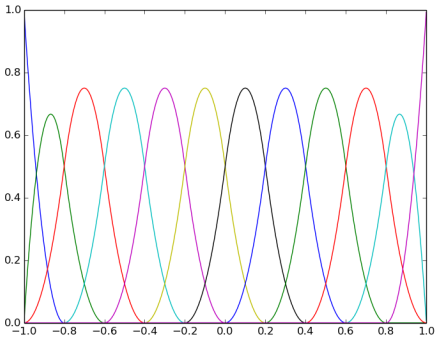
- Partition of unity: The sum of the B-spline basis functions satisfy partition of unity i.e.

$$\sum_{i=0}^{n} N_{i,p}(\mathbf{t}) = 1$$

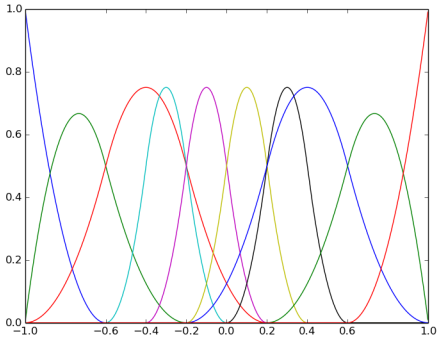
To illustrate these properties, 1D univariate B-spline basis are used. Figure 4.1 shows B-spline polynomials that are linear, quadratic and cubic respectively. These are generated from an uniformly spaced knot vectors with end knots of -1. and 1. and multiplicity of $p+1$. It is obvious to note that both the positivity and local support



(a) Coarse grid quadratic B-spline



(b) Fine grid quadratic B-spline



(c) HB-spline Basis as a combination of fine and coarse grid basis functions

Figure 4.2: Construction of an univariate Quadratic HB-spline. The region [-0.6, 0.6] is selected for refinement on the coarse grid and replaced with the 4 corresponding B-spline basis from the refined grid.

property is satisfied by the B-spline polynomials. Also, partition of unity is satisfied and the contribution of the number of B-spline polynomials at each point along the knot space is $p+1$. The smoothness property is also verified by comparing the linear and quadratic B-spline. For the linear B-spline, it is 0 times smoothly differentiable which is obvious from the non-smooth curves while for the quadratic B-spline, it is 1 time differentiable and the differentiated polynomial will be similar to that of the linear case.

Hierarchical B-splines (HB-splines) enable the use of local mesh refinement with B-spline basis functions [140]. An illustrative 1D example is shown in Figure 4.2. Figure 4.2 illustrates a coarse grid quadratic B-spline basis corresponding to a knot vector

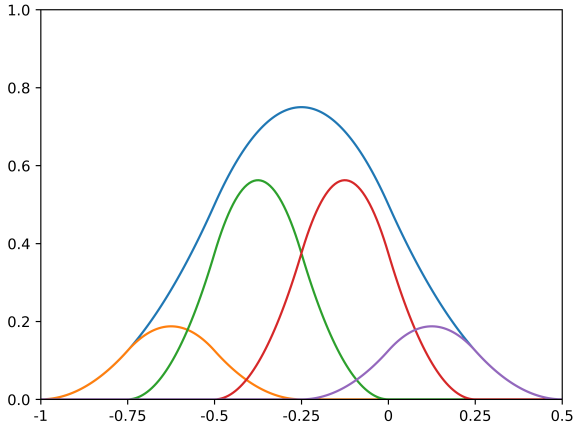
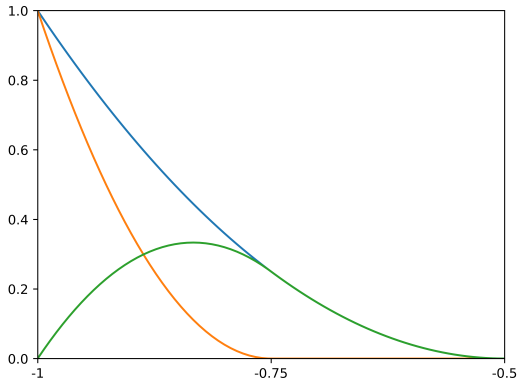


Figure 4.3: The quadratic B-spline on the coarse grid (blue curve) can be represented by a weighted linear combination of its 4 children from the fine grid.

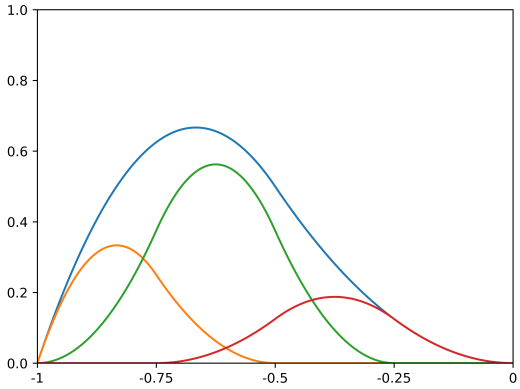
$t = \{-1, -1, -1, -0.6, -0.2, 0.2, 0.6, 1, 1, 1\}$ and a fine grid quadratic B-spline basis corresponding to knot vector $t_{ref} = \{-1, -1, -1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1\}$ which is an uniform refinement of the coarse grid knot vector. If the interval $[-0.6, 0.6]$ is chosen for refinement, any B-spline basis function that has its support contained entirely in this region is also refined. In this case, only one curve in the coarse grid is refined and the rest of the functions are unaffected. Using the fine grid knot vector, four basis functions corresponding to the same region can be substituted into the coarse grid when the interval is locally refined as shown in Figure 4.2c. This is valid as basis functions with grid size h are algebraically related to the basis functions with grid size $\frac{h}{2}$ as follows:

$$N_{i,p}^{l}(\mathbf{t}) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_{2i+j,p}^{l+1}(\mathbf{t}), \tag{4.4}$$

Equation (4.4) relates the basis function from a coarse grid level l to a fine grid level $l+1$, assuming the fine grid is an uniform refinement of the coarse grid level, through the subdivision approach. Figure 4.3 shows the linear combination from the fine grid that is required to form the corresponding basis function in the coarse grid from Figure 4.2. Additionally, more expressions can be generated if refinement of the



(a) Representation of the outermost B-spline on the coarse grid (blue curve) on fine grid



(b) Representation of the 2nd outermost B-spline on the coarse grid (blue curve) on fine grid

Figure 4.4: Refinement of the 2 quadratic B-spline polynomial that does not span 3 knot spaces.

B-spline polynomials that do not span $p+1$ knot span is desired. These expressions are not easily generalized but can be easily derived as part of the subdivision matrix. More details are shown in Section 4.2. For a quadratic B-spline polynomial, the refinement of the two polynomials on the coarse-grid can be represented as such It is straightforward to observe from Figures 4.3 and 4.4 that partition of unity is not preserved. Therefore, to integrate this formulation within a FEM framework, projection and scaling of the HB-spline polynomials are required [141].

THB-splines offer several advantages over HB-splines. A distinct advantage is that after refinement, THB-spline basis functions retain the partition of unity property, making them compatible with interpolations of the density field. For THB-splines, the construction of the coarse and fine grid B-spline basis functions remains unchanged and the refined region has its basis functions from the coarse grid replaced by the corresponding basis functions in the fine grid. However, the neighboring B-spline that shares the same children B-splines as the refined basis function in the refinement region must undergo truncation. During this truncation, the truncated neighboring basis function is computed by a linear combination of its passive children B-spline from the finer grid, causing the magnitude of basis function that is within the refinement

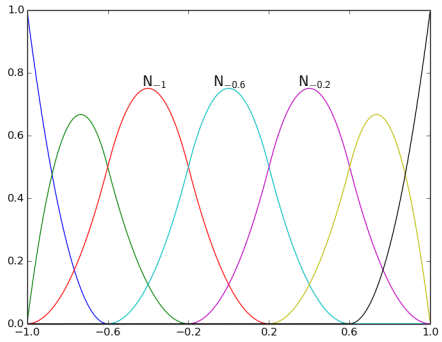
Table 4.1: Parent B-spline basis function from Figure 4.5a and its corresponding children B-spline from Figure 4.5b

Parent B-spline	Corresponding Children
N_{-1}	M_{-1},$M_{-0.8}$,$M_{-0.6}$,$M_{-0.4}$
$N_{-0.6}$	$M_{-0.6}$,$M_{-0.4}$,$M_{-0.2}$,M_0
$N_{-0.2}$	$M_{-0.2}$,M_0,$M_{0.2}$,$M_{0.4}$

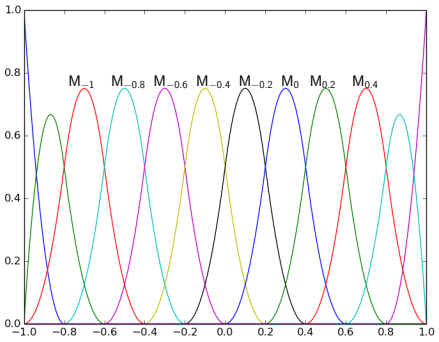
region to be reduced as shown in Figure 4.5c. In addition, the support interval of these neighboring B-spline is shortened since the passive children B-spline do not span over the original support domain.

From Figure 4.5a, the basis functions of interest are N_{-1}, $N_{-0.6}$, $N_{-0.2}$. Their corresponding children B-spline functions on the finer grid are labeled on Figure 4.5b and the parent-children B-spline relationship are tabulated on Table 4.1. $N_{-0.6}$ is the only B-spline that undergoes refinement in the domain $[-0.6, 0.6]$ and it is replaced by its four children B-spline from the finer grid as shown previously. However, since M_1 and M_3 shared the same children B-spline $M_{-0.6}$,$M_{-0.4}$ and $M_{-0.2}$,M_0 respectively, they are truncated. The truncated N_{-1} and $N_{-0.6}$ are computed by a linear combination of its passive children, i.e. not involved in the refinement of $N_{-0.6}$, which are M_{-1}, $M_{-0.8}$ and $M_{0.2}$,$M_{0.4}$ respectively. Therefore the truncated neighboring B-spline have its support reduced from the initial $[-1, 0.2]$ to $[-1, -0.2]$ as well as from $[-0.2, 1]$ to $[0.2, 1]$. Comparing Figure 4.2c and 4.5d, there are fewer overlapping THB-spline function compared to HB-spline for the same region of refinement, making the resulting linear system sparser and less computational resources are needed to solve it. 2D/3D THB-spline basis functions can be constructed using the univariate 1D THB-spline basis functions and its corresponding knot vectors through tensor product.

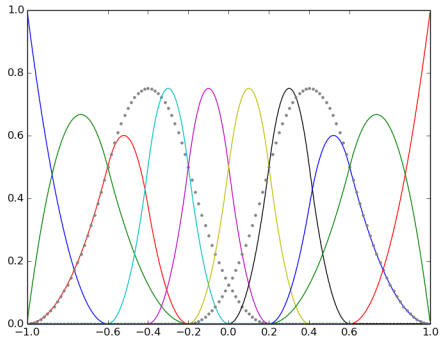
An issue with integrating THB-spline with FEM applications is the inconsistent number of active shape functions across elements as seen in Figure 4.5d. This is due



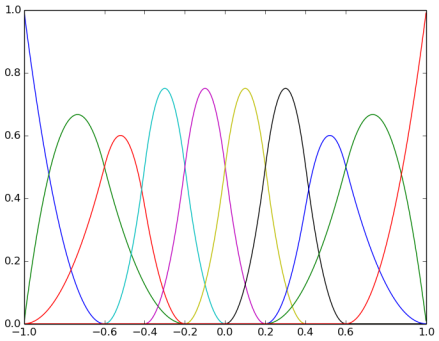
(a) Coarse Grid Quadratic B-spline



(b) Fine Grid Quadratic B-spline

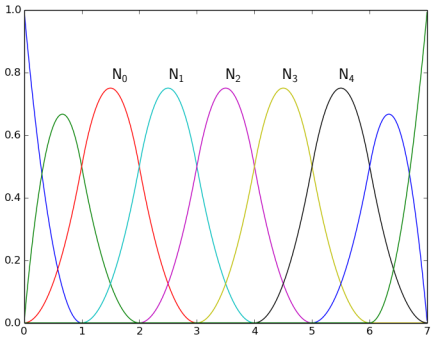


(c) The truncation of the neighboring coarse grid B-spline outlined in grey

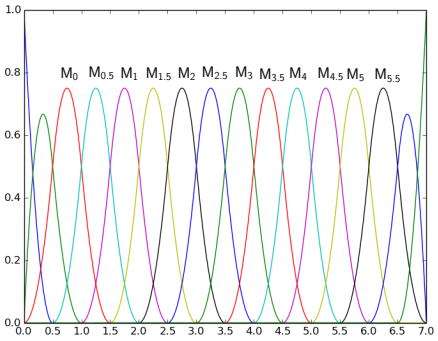


(d) THB-spline basis represented on the coarse grid level

Figure 4.5: Construction of an univariate Quadratic THB-spline. The region $[-0.6, 0.6]$ is selected for refinement on the coarse grid and replaced with the corresponding B-spline basis from the refined grid. The neighboring B-spline on the coarse grid that is nonzero in the interval is truncated



(a) Coarse Grid Quadratic B-spline



(b) Fine Grid Quadratic B-spline

Figure 4.6: Parent B-spline and its corresponding children B-spline.

to the truncation of B-spline functions across grid levels. As mentioned previously, truncation imposed in the form of external constraints on the B-spline functions can alleviate this concern. The following example demonstrates truncation through constraints. Figure 4.6 shows the coarse and fine grid B-spline polynomials. When N_2, with support $[2.0, 5.0]$, undergoes uniform refinement as shown in Figure 4.7, the coarse grid-fine grid B-spline relation, as presented in (4.4), can be explicitly stated as

$$\begin{aligned}
\mathrm{N}_0 &= \frac{1}{4}\mathrm{M}_0 + \frac{3}{4}\mathrm{M}_{0.5} + \frac{3}{4}\mathrm{M}_1 + \frac{1}{4}\mathrm{M}_{1.5}, \\
\mathrm{N}_1 &= \frac{1}{4}\mathrm{M}_1 + \frac{3}{4}\mathrm{M}_{1.5} + \frac{3}{4}\mathrm{M}_2 + \frac{1}{4}\mathrm{M}_{2.5}, \\
\mathrm{N}_2 &= \frac{1}{4}\mathrm{M}_2 + \frac{3}{4}\mathrm{M}_{2.5} + \frac{3}{4}\mathrm{M}_3 + \frac{1}{4}\mathrm{M}_{3.5}, \\
\mathrm{N}_3 &= \frac{1}{4}\mathrm{M}_3 + \frac{3}{4}\mathrm{M}_{3.5} + \frac{3}{4}\mathrm{M}_4 + \frac{1}{4}\mathrm{M}_{4.5}, \\
\mathrm{N}_4 &= \frac{1}{4}\mathrm{M}_4 + \frac{3}{4}\mathrm{M}_{4.5} + \frac{3}{4}\mathrm{M}_5 + \frac{1}{4}\mathrm{M}_{5.5},
\end{aligned} \tag{4.5}$$

Since N_1 and N_3 shared children B-spline M_2, $\mathrm{M}_{2.5}$ and M_3, $\mathrm{M}_{3.5}$ of N_2 respectively, they undergo truncation. The truncated basis functions of N_1 and N_3, N_1' and N_3' respectively, can be expressed as

$$\begin{aligned}
\mathrm{N}_1' &= \frac{1}{4}\mathrm{M}_1 + \frac{3}{4}\mathrm{M}_{1.5}, \\
\mathrm{N}_3' &= \frac{3}{4}\mathrm{M}_4 + \frac{1}{4}\mathrm{M}_{4.5},
\end{aligned} \tag{4.6}$$

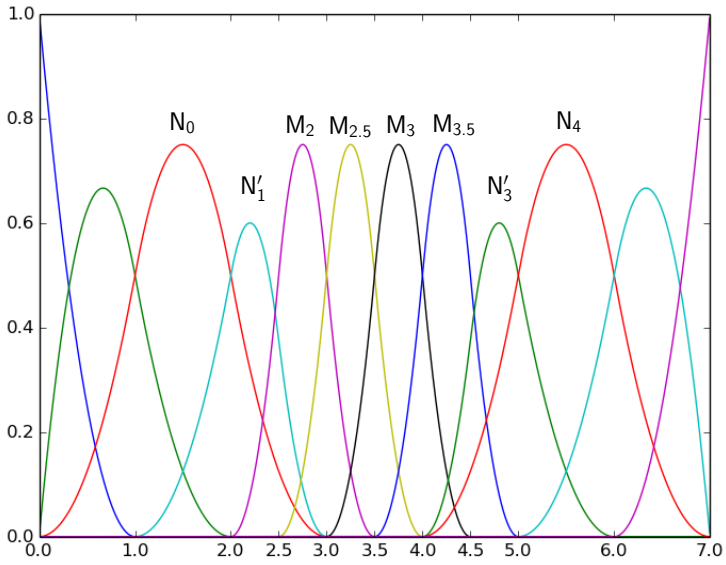


Figure 4.7: Basis space with truncation of N_2

Over the element $[1.0, 2.0]$, the truncated N'_1 is active and its expression in terms of the children B-spline differs from N_1, as shown in (4.5). However, since M_2 and $M_{2.5}$ are zero over this interval, they have no contribution towards the non-truncated basis function N_1 and N_1 and N'_1 are equivalent in $[1.0, 2.0]$. Similarly, N_3 and N'_3 are equivalent in $[5.0, 6.0]$. For the refined element $[2.0, 2.5]$, the active B-spline functions are N_0, N'_1 and M_2. Instead of working explicitly with B-spline functions from different grid level, they can be expressed in terms of their children B-spline functions, M_1, $M'_{1.5}$ and M_2, as shown in (4.7).

$$\begin{aligned} N_0 &= \frac{3}{4}M_1 + \frac{1}{4}M_{1.5}, \\ N'_1 &= \frac{1}{4}M_1 + \frac{3}{4}M_{1.5}, \end{aligned} \tag{4.7}$$

This approach is similar to the subdivision project vector method proposed by Bornemann and Cirak [142], but a major difference is that the linear combination of the children B-spline functions can be posed as sparse constraints, which can be handled efficiently through the topology optimization framework. This keeps the modular

nature of the FEM approach intact. Expressing the spline curve, u, for the element $[2.0, 2.5]$ in terms of its active B-spline polynomials,

$$\begin{aligned} \mathrm{u} &= a_0 \mathrm{N}_0 + a_1 \mathrm{N}_1' + b_2 \mathrm{M}_2 \\ &= a_0 \left(\frac{3}{4} \mathrm{M}_1 + \frac{1}{4} \mathrm{M}_{1.5} \right) + a_1 \left(\frac{1}{4} \mathrm{M}_1 + \frac{3}{4} \mathrm{M}_{1.5} \right) + b_2 \mathrm{M}_2 \\ &= \left(\frac{3}{4} a_0 + \frac{1}{4} a_1 \right) \mathrm{M}_1 + \left(\frac{1}{4} a_0 + \frac{3}{4} a_1 \right) \mathrm{M}_{1.5} + b_2 \mathrm{M}_2 \\ &= b_1 \mathrm{M}_1 + b_{1.5} \mathrm{M}_{1.5} + b_2 \mathrm{M}_2, \end{aligned} \tag{4.8}$$

Since u can be expressed either in terms of $[\mathrm{M}_1, \mathrm{M}_{1.5}, \mathrm{M}_2]$ or $[\mathrm{N}_0, \mathrm{N}_1', \mathrm{M}_2]$ and they must be identical, two constraints associated with the coefficients of M_0, M_1 are imposed (4.9)

$$\begin{aligned} b_1 &= \frac{3}{4} a_0 + \frac{1}{4} a_1, \\ b_{1.5} &= \frac{1}{4} a_0 + \frac{3}{4} a_1, \end{aligned} \tag{4.9}$$

Similarly for the refined element $[2.5, 3.0]$, the active B-spline polynomials can be expressed in terms of the refined level B-spline polynomials M_i

$$\begin{aligned} \mathrm{N}_0 &= \frac{1}{4} \mathrm{M}_{1.5}, \\ \mathrm{N}_1' &= \frac{3}{4} \mathrm{M}_{1.5}, \end{aligned} \tag{4.10}$$

Expressing the spline curve, u, for the element $[2.5, 3.0]$ in terms of its active B-spline polynomials

$$\begin{aligned} \mathrm{u} &= a_0 \mathrm{N}_0 + a_1 \mathrm{N}_1' + b_2 \mathrm{M}_2 + b_{2.5} \mathrm{M}_{2.5} \\ &= \left(\frac{1}{4} a_0 + \frac{3}{4} a_1 \right) \mathrm{M}_{1.5} + b_2 \mathrm{M}_2 + b_{2.5} \mathrm{M}_{2.5} \\ &= b_{1.5} \mathrm{M}_{1.5} + b_2 \mathrm{M}_2 + b_{2.5} \mathrm{M}_{2.5}, \end{aligned} \tag{4.11}$$

For this element, only one constraint is required

$$b_{1.5} = \frac{1}{4} a_0 + \frac{3}{4} a_1, \tag{4.12}$$

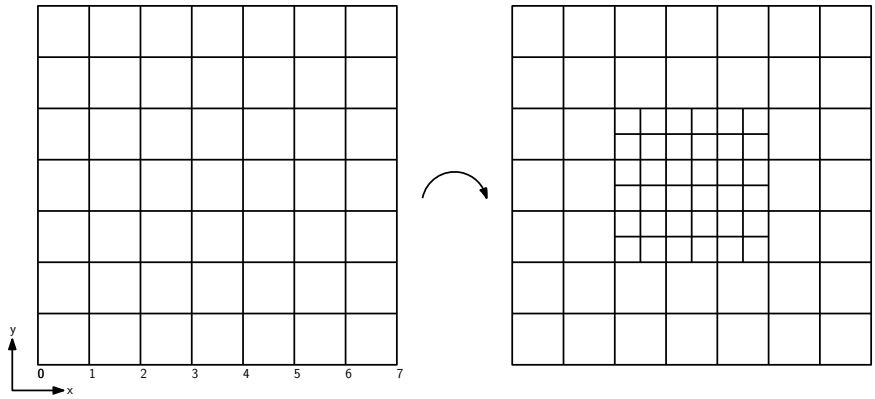


Figure 4.8: Refinement of N_2 in both x and y for a 2D case

As mentioned earlier in this section, using the constraint approach simplifies the integration of THB-spline basis polynomials with existing FEM framework since the number of active B-spline functions over the element is consistent with that of Lagrange basis functions.

For higher dimensions, the formulation of the constraints take a similar approach since they can be represented by tensor products of a sequence of 1D curves. Figure 4.8 shows the analogous refinement of M_1 for both directions in the form of a 2D grid. By refining N_2 in both the x and y direction, the resulting biquadratic tensor product $N_{2x}N_{2y}$ is replaced by 16 children B-spline tensor product. In this example, N_{1x}, N_{3x}, N_{1y} and N_{3y} are truncated and constraints can be imposed in certain refined elements. The number of constraints in the refined domain depends on its location, ranging from 3 to a maximum of 8. Figure 4.9 shows the number of constraints in the refined domain and how it differs within the region. The 2D constraint derivation for the different region are detailed in Appendix A.1. While mathematically there are a total of 180 constraints in this simple refinement example, the number of constraints required for implementation is much fewer due to overlapping active constraints in each region. The number of unique, sparse linear constraints is reduced from 180 to 48.

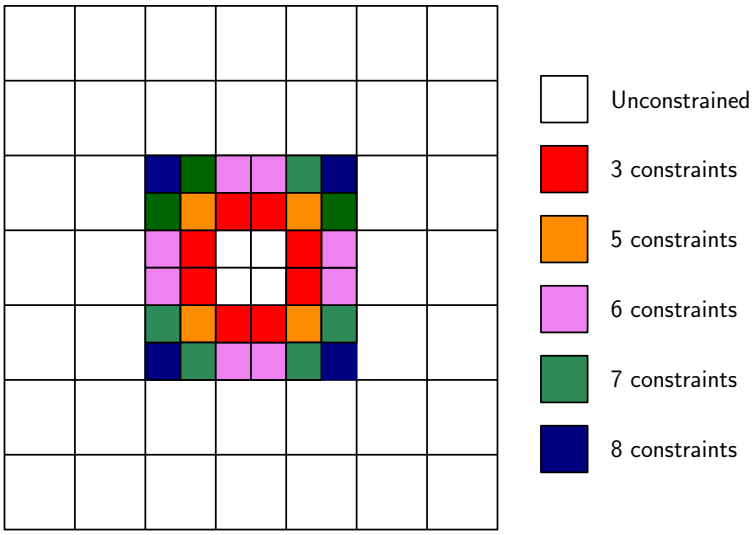


Figure 4.9: Differing number of constraints in the domain denoted by the different color, with a total of 180 mathematical constraints

2D THB-spline Topology Optimization Results

In this section, compliance-minimization, mass constrained results for topology optimization using THB-spline FEM are presented. Quadratic B-spline polynomials are used to for both the analysis and design parametrization. Adaptive mesh refinement is used unless specified. Note that to derive these results, the design variable is assigned to each element in the analysis mesh and a spatial filter that resembles the bi-quadratic B-spline polynomial is used. This is unlike the nodal filter parametrization used in most of this work. A number of 2D examples are investigated and are detailed as follows. The first example is a common topology optimization problem which is the MBB beam. Figure 4.10 shows the domain and loading conditions on the MBB beam. For this problem, the beam is taken to be 1 cm thick, simply supported at both ends and loaded with point load P set at 1 kN. L is given to be 1.0 m. There are 12 000 elements in the discretized domain. The mass constraint is kept at 25%. No adaptive mesh refinement is used for this design. Figure 4.11 shows the resulting MBB design. The resulting MBB design resembles what is commonly seen in literature. Due to the use of the element-wise design variable, the transition from

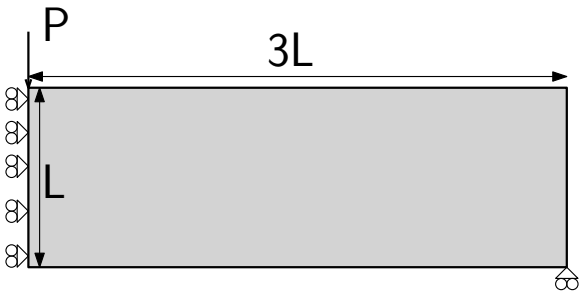


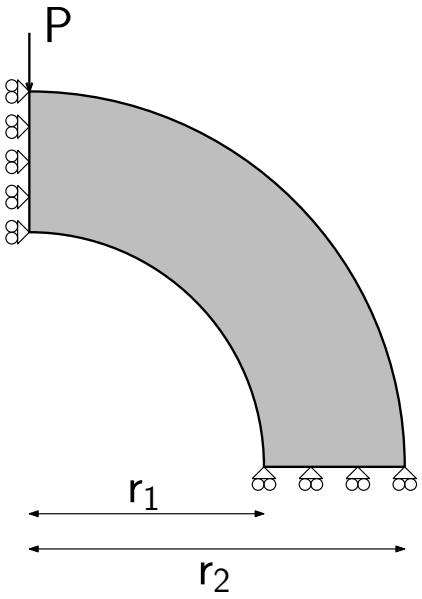
Figure 4.10: Schematic of a MBB beam

Figure 4.11: Compliance minimization design using quadratic B-spline basis functions

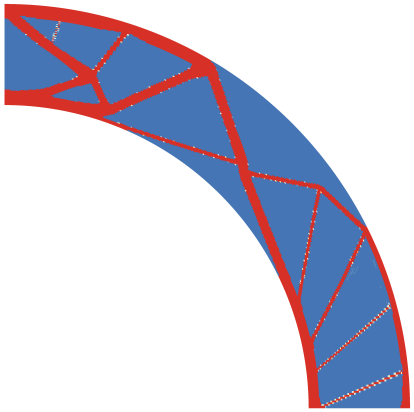
solid to void is more abrupt. Also, since adaptive mesh refinement is not used, the boundaries of the design is less smooth.

The second 2D example deals with a variation of the MBB beam. As shown in Figure 4.12a, the beam is curved but the domain is loaded similarly to the MBB beam. For this problem, the beam is taken to be 1 cm thick, simply supported at both ends and loaded with point load P set at 1 kN. r_1 is set at 3.0 m and r_2 is set at 4.0 m. The domain is discretized into 30 000 elements. The mass constraint is kept at 25%. Similar to the MBB problem, no adaptive mesh refinement is used. Figure 4.12b shows the resulting curvilinear beam design. The resulting design has many cross-link trusses that extends from one end of the domain to the other to maximize the stiffness. The cross-link truss which runs almost parallel to the load is also the thickest since it improves both axial and bending stiffness. This also account for the thinner members on the bottom right of the domain as they do not carry as much bending load.

The next 2D example investigated is a variation of the serpentine beam that was used in [143]. The domain is shown in Figure 4.13a. For this problem, the serpentine beam is taken to be 1 cm thick, with one end clamped and loaded with point load



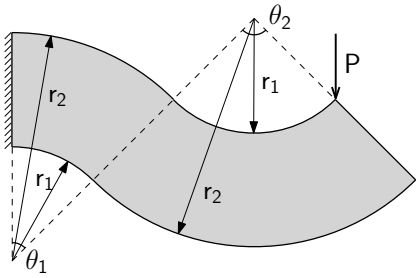
(a) Schematic of a curvilinear beam



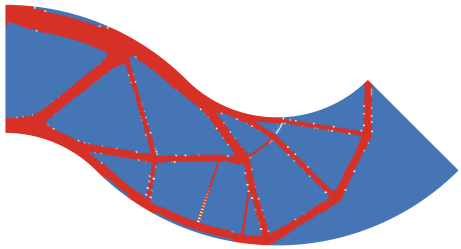
(b) Compliance minimization design

Figure 4.12: Diagram and compliance design of curvilinear beam using quadratic B-spline polynomials

P = 500 N. r_1 is set at 1.0 m and r_2 is set at 2.0 m. θ_1 is set at 45°and θ_2 is set at 90.0°. The domain is discretized into 24 000 elements and the mass constraint is kept at 25%. For this case, no adaptive mesh refinement is used. The resulting design is shown in Figure 4.13b. The design in Figure 4.13b displays many traits that are seen Figure 4.12b, with a main truss extending from the load application point to the cantilevered end. Due to the curved geometry, the trusses formed on both sides of the cantilevered root carry the resulting axial load and bending moment, with smaller



(a) Schematic of a serpentine beam



(b) Compliance minimization design using quadratic B-spline basis functions

Figure 4.13: Diagram and compliance design of serpentine beam

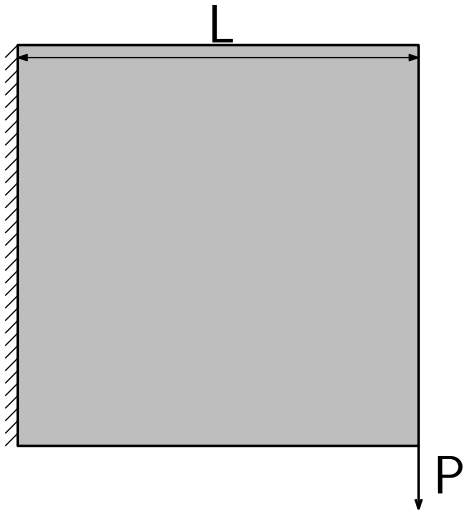
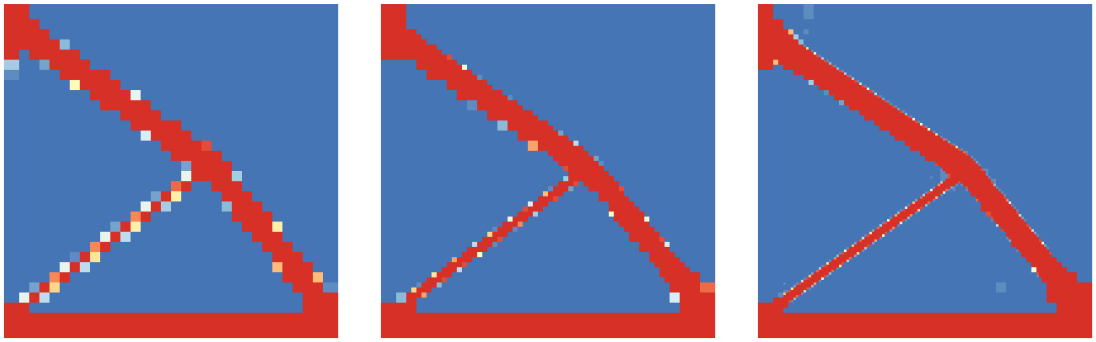


Figure 4.14: Schematic of a cantilever plate

trusses connecting them together for rigidity.

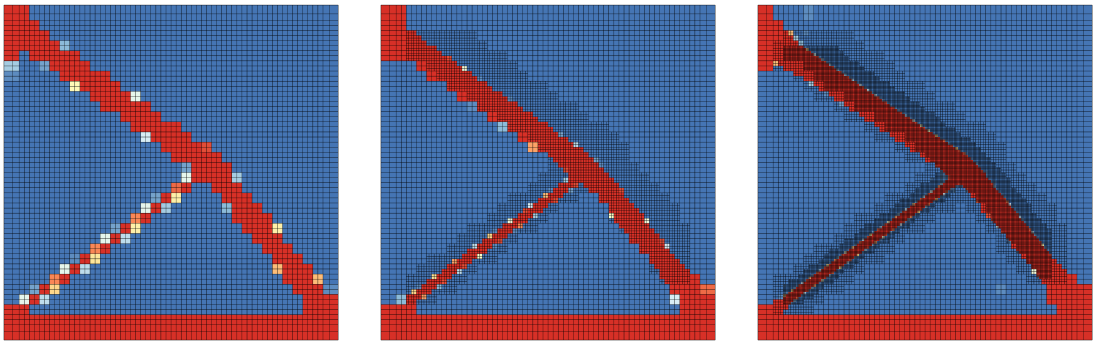
The last example investigated is a square cantilever plate domain shown in Figure 4.14. For this problem, the plate is taken to be 1 cm thick, clamped at one end and loaded with point load P set at 1 kN. L is given to be 5.0 m. The initial mesh has 32×32 elements, a total of 1024 elements. The mass constraint is fixed at 25%. Two cycles of adaptive mesh refinement is used and unlike previous cases, the mesh refinement involves refining but not coarsening. The designs are adaptively refined after every 250 iterations. After the first refinement, there are 1819 elements in the mesh and after the second refinement, there are 4036 elements. Figure 4.15 shows the resulting plate design and its underlying mesh at each cycle. It is noticeable that the evolution of the design behaves similar to adaptive mesh refinement when used with Lagrange basis functions. The initial design in Figure 4.15a has many intermediate densities present, as expected. As the cross-links are adaptively refined as shown in Figures 4.15b and 4.15c, a nearly discrete 0-1 design is formed and it closely resembles what is shown in literature. The adaptive mesh refinement process also created a structural layout with smoother boundary as the grid size becomes finer.



(a) Design before adaptive mesh refinement

(b) Design after one cycle of adaptive mesh refinement

(c) Design after two cycles of adaptive mesh refinement



(d) Mesh before adaptive mesh refinement

(e) Mesh after one cycle of adaptive mesh refinement

(f) Mesh after two cycles of adaptive mesh refinement

Figure 4.15: Design of plate using THB-spline basis functions at each refinement cycle

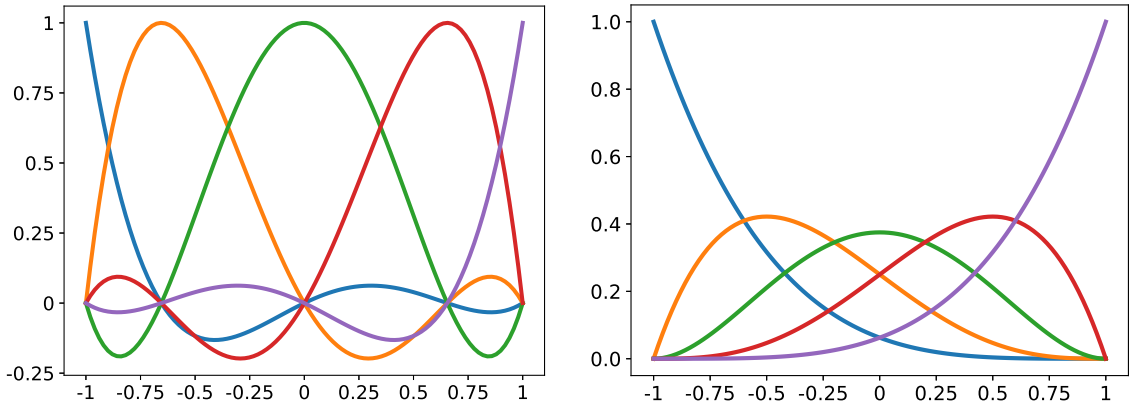
Higher Order Design Parametrization for Thermoelastic Topology Optimization

In this section, I describe the efforts made to utilize higher order parametrization techniques for both analysis and design within the multimaterial thermoelastic topology optimization framework. For analysis parametrization, higher order Lagrange polynomials are suitable since there is no restriction on the sign of the basis function. However, for design parametrization, a suitable higher order formulation should be non-negative everywhere, so that non-physical negative design densities are avoided. Ideally, the design density basis should also satisfy the partition of unity property so that addition scaling of the polynomials is not required. For design parametrization, higher order Lagrange polynomials are not suitable, as they can be negative, as shown in Figure 4.16a. On the other hand, Bernstein polynomials, as shown in Figure 4.16b, are non-negative everywhere and satisfy a partition of unity. Bernstein polynomials are a subset of B-spline polynomials and have been used for density parametrization for 2D structural topology optimization [95]. Bernstein polynomial can be expressed explicitly or implicitly. An explicit expression the Bernstein basis is as follows:

$$N_{i,p} = \binom{p}{i}(1-t)^{p-k}t^k, \tag{4.13}$$

for $t \in [0,1]$. For an arbitrary knot space $t \in [t_1, t_2]$, the Bernstein polynomial can be implicitly evaluated using the knot vector $t = \{t_1, t_1, ..., t_1, t_2, t_2, ..., t_2\}$, where there are p entries of t_1 and t_2. Bernstein polynomials share certain properties with B-spline polynomials shown in Section 4.1. The critical properties of Bernstein polynomials are summarized as follows:

- Positivity: The pth order Bernstein polynomial $N_{i,p}$ is non-negative
- Partition of unity: The sum of the Bernstein polynomial satisfy partition of



(a) 5th order Lagrange polynomial

(b) 5th order Bernstein polynomial

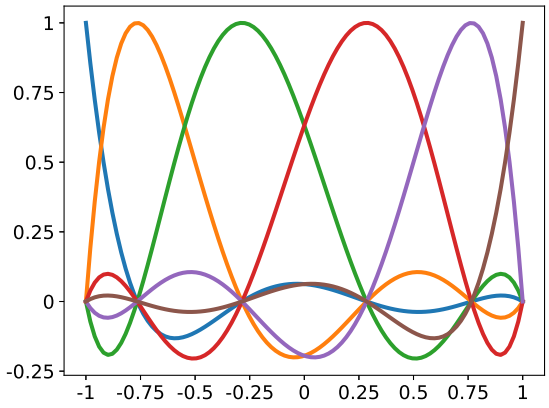
Figure 4.16: Differences between univariate Lagrange polynomial and Bernstein polynomial

unity i.e.

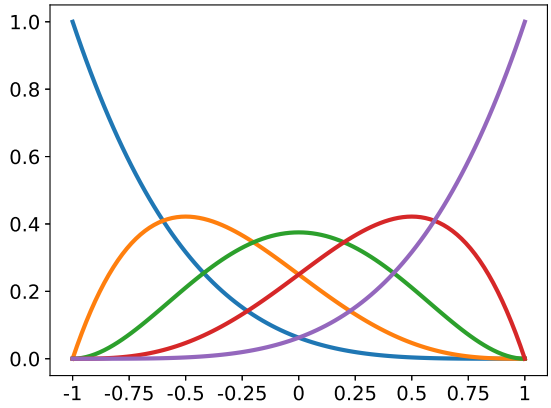
$$\sum_{i=0}^{n} N_{i,p}(\mathbf{t}) = 1$$

Figure 4.16 illustrates these properties as well as the difference between a 5th order Bernstein polynomial and 5th order Lagrange polynomial using a 1D univariate example. As mentioned above, higher order Lagrange polynomials can be negative and this can lead to non-physical interpolated density values. More details on the Bernstein polynomial and its application can be found in [144]. In this work, an investigation involving a series of analysis and design mesh orders to analyze the effects on the topology optimization results and convergence is carried out. Figure 4.17 shows the corresponding 1D univariate example with 6th order analysis and 5th order design parametrization.

In order to use Bernstein polynomials within this framework, expressions have to be developed in order to interpolate between meshes that have different mesh spacing due to local mesh refinement, as well as meshes that differ in terms of its polynomial order, for geometric multigrid. The mesh interpolation relationship is dependent on the order of the Bernstein polynomial as well as mesh grid spacing, and it can be evaluated analytically through subdivision. This can be represented by the following



(a) 6th order Lagrange polynomial for analysis



(b) 5th order Bernstein polynomial for design

Figure 4.17: Univariate 6th order Lagrange polynomial and 5th order Bernstein polynomial for analysis and design respectively.

relationship:

$$\mathbf{N^c}_p = \mathbf{S}_{p,q}\mathbf{N^f}_q \tag{4.14}$$

where p and q are the orders of the Bernstein polynomial, $\mathbf{N^c}_p$ and $\mathbf{N^f}_p$ are the Bernstein polynomial on the two different meshes, respectively, and $\mathbf{S}_{p,q}$ is the subdivision matrix. This is important when handling adaptive mesh refinement where dependent nodes have to be constrained, as well as when interpolating across meshes of different orders for use in geometric multigrid. While the constraints for the dependent nodes can be accounted for through the optimizer, as shown in the work of Lambe and Czekanski [95], for present work, the constraints are enforced internally through the framework. To develop the constraints for the dependent nodes, the univariate spline curve u can be expressed as [137, 142]:

$$\begin{aligned} u &= \mathbf{N^c}_p\mathbf{u^c}_p \\ &= \left(\mathbf{S}_{p,q}\mathbf{N^f}_q\right)\mathbf{u^c}_p \\ &= \mathbf{N^f}_q\left(\mathbf{S}^T_{p,q}\mathbf{u^c}_p\right) \\ &= \mathbf{N^f}_q\mathbf{u^f}_q \end{aligned} \tag{4.15}$$

where $\mathbf{u^c}_p$ and $\mathbf{u^f}_q$ are the control points on the two different meshes respectively so

$$\mathbf{u^f}_q = \mathbf{S}^T_{p,q}\mathbf{u^c}_p \tag{4.16}$$

Thus, the weights of the transpose of the subdivision matrix serve as the constraints as the meshes are interpolated across mesh order and mesh grid size. First, I address the subdivision matrix formulation for the case when the mesh is adaptively refined by adding new independent and dependent nodes. To interpolate a mesh with polynomial order $p = 5$ and mesh grid spacing h to a mesh that has polynomial order $q = 5$ and half the mesh grid spacing i.e. $\frac{h}{2}$, the following expression holds

$$\begin{bmatrix} \mathrm{N}^c_0 \\ \mathrm{N}^c_1 \\ \mathrm{N}^c_2 \\ \mathrm{N}^c_3 \\ \mathrm{N}^c_4 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{8} & \frac{1}{4} & \frac{1}{8} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{8} & \frac{1}{4} & \frac{3}{8} & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}^f_0 \\ \mathrm{N}^f_1 \\ \mathrm{N}^f_2 \\ \mathrm{N}^f_3 \\ \mathrm{N}^f_4 \\ \mathrm{N}^f_5 \\ \mathrm{N}^f_6 \\ \mathrm{N}^f_7 \\ \mathrm{N}^f_8 \end{bmatrix}, \tag{4.17}$$

Figure 4.18 shows how the Bernstein polynomial on the coarse mesh can be represented by a weighted linear combination of Bernstein polynomials on the fine mesh, with the weights expressed through the subdivision matrix 4.17. Additional expressions derived for other values of p and q are in Appendix A.2. An interesting note in the evaluation of the weights of this subdivision matrix $\mathbf{S}_{p,q}$ is that the nonzero values in the columns do not change as the polynomial order increases and the nonzero weights of pth column can be computed by evaluating the midpoint of the pth or-

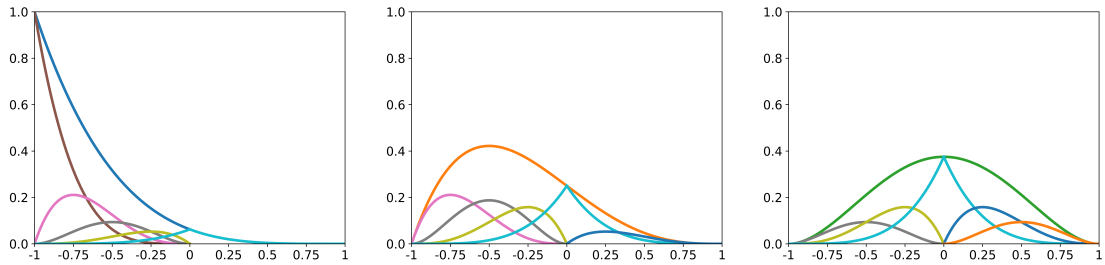


Figure 4.18: Subdivision expression across meshes with half the mesh spacing of the other. In this case, the polynomial on the coarse mesh (blue curve) can be represented by a linear combination of 5 polynomials on the fine mesh which has half the mesh spacing as the coarse mesh.

der Bernstein polynomial. Thus, the constraints on the dependent nodes when using Bernstein polynomial of pth order can be easily computed. The constraints derived for the dependent nodes here are for 2D domains. The extension of these expressions to 3D is straightforward and it is through the multivariate tensor product formulation.

As mentioned above, the subdivision matrix can also be derived for interpolating across meshes of different orders. For this work, it is used for interpolating between the hierarchy of meshes generated for geometric multigrid. Multigrid is required in order to solve large-scale systems efficiently, especially in 3D. For instance, to interpolate from a mesh with polynomial order $p = 4$ to a mesh with polynomial order $q = 5$ and both meshes having identical mesh grid spacing,

$$\begin{bmatrix} \mathrm{N}_0^c \\ \mathrm{N}_1^c \\ \mathrm{N}_2^c \\ \mathrm{N}_3^c \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{3}{4} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{3}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}_0^f \\ \mathrm{N}_1^f \\ \mathrm{N}_2^f \\ \mathrm{N}_3^f \\ \mathrm{N}_4^f \end{bmatrix}, \tag{4.18}$$

Similarly, Figure 4.19 shows how the Bernstein polynomial on a lower order mesh can be represented by a weighted linear combination of Bernstein polynomials on a higher order mesh, with the weights expressed through the subdivision matrix 4.18. This is

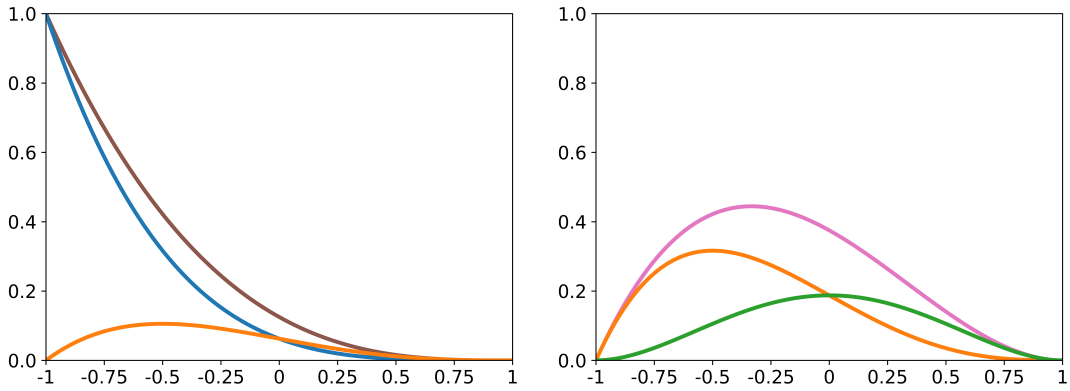


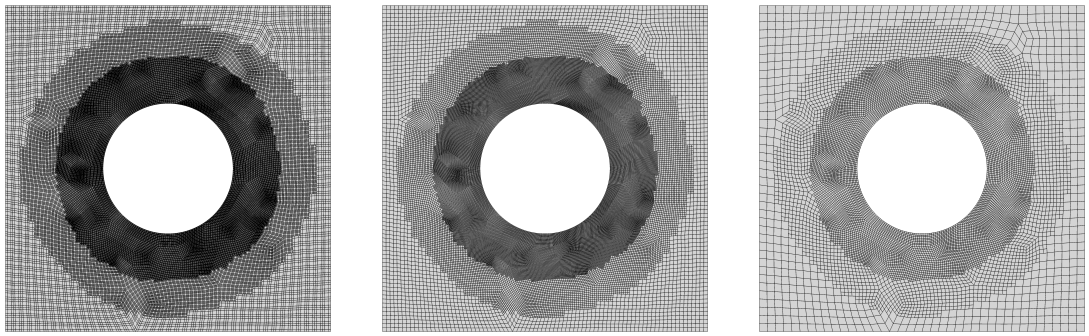
Figure 4.19: Subdivision expression across meshes with different polynomial orders. In this case, the polynomial (blue curve) on the mesh with order $p = 4$ can be represented by a linear combination of 2 polynomials on the higher order mesh $q = 5$

valid for a 2D domain and the extension of these expressions to 3D is through the multivariate tensor product formulation. Additional expressions for different values of p and q are shown in Appendix A.2. While the expressions derived here are assuming that either the polynomial order of the two meshes are identical or the mesh grid spacing across meshes are equivalent, subdivision expressions for interpolating across meshes with different orders and mesh grid spacing is straightforward since this unknown subdivision matrix can be evaluated as a product of a combination of the subdivision matrices derived here.

Solution Methods for Higher Order Parametrization

While higher order analysis and parametrization can be used within the existing topology optimization framework for elements with Lagrange-type basis functions, a few minor modifications to the solution method can be used to improve numerical performance. Similar to the work presented in Chapter 3, a node-based parametrization [130] is used in order to have efficient and consistent design representation between meshes with different levels of local refinement.

The first modification is in the evaluation of the design mesh. The design field is evaluated on a mesh that is an order lower than that of the analysis mesh for the



(a) 4th order Gauss-Lobatto analysis mesh

(b) 3rd order Gauss-Lobatto medium mesh

(c) 2nd order Gauss-Lobatto mesh at coarsest level

Figure 4.20: Illustration of the construction of the meshes for geometric multigrid. The analysis mesh is the first geometric multigrid level and the subsequent geometric multigrid levels are obtained by reducing the mesh order of the previous mesh through TMR.

purpose of removing numerical artifacts [88]. Similar to Section 3.1, each node in the design mesh is associated with $M+1$ design variable values $\mathbf{x}$. However, instead of using a bilinear/trilinear interpolation as shown in Equation (3.5) to obtain the design variable values associated with each element in the analysis mesh, the higher order Bernstein polynomial interpolation allows for the design variable values to vary across the element, thus providing greater design freedom. A node-based parametrization also enables the use of geometric multigrid solution strategies. For multigrid mesh generation, the hierarchy of meshes are created using the quadtree/octree approach through TMR as described above. An illustrative example is shown in in Figure 4.20. Each coarse mesh level is created automatically by successive reduction of the mesh order of the fine, adaptively refined analysis mesh which, in this case, is a 4th order Gauss-Lobatto mesh. Another modification of the numerical solution scheme for numerical performance is through the multigrid preconditioner itself. At each optimization iteration, the large linear system of governing equations arising from the finite-element discretization is solved using a parallel geometric multigrid-preconditioned Krylov subspace method [27]. On all but the coarsest mesh, a block

symmetric Gauss–Seidel smoothing operation is used to attenuate the high frequency solution error. In this work, GMRES(50) with a maximum of five restarts is used to solve the governing equations [109]. Also, based on numerical experiments, I found that convergence on the geometric multigrid-preconditioned Krylov subspace method for large-scale 3D examples can be improved by using an under-relaxation coefficient of 0.5 that is associated with the block Gauss-Seidel smoothing operation. Lastly, the feature-based adaptive refinement heuristics is slightly modified so that the values of x_{rl} and x_{ru} are chosen to be $x_{rl} = 0.95$ and $x_{ru} = 0.90$ respectively. Also, for the final design to be considered as converged, the following criterion is used

$$\frac{f_{k-1} - f_k}{f_{k-1}} < \gamma \tag{4.19}$$

where f_k is the objective value at the kth iteration and γ is a constant parameter that denotes the relative tolerance. For this work, $\gamma = 10^{-4}$ unless specified otherwise.

Results

Higher order analysis and design parametrization developed are used to create multi-material thermoelastic topology optimization designs. The focus in this section is on the mass-constrained compliance-minimization problem, with the formulation shown in (3.24). Penalization parameters are set to $q_D = 8.0$, $q_\lambda = 0.0$, and $q_\beta = 8.0$, respectively. For both the 2D and 3D problems investigated, there are two candidate materials and their properties are listed in Table 4.2.

The first example is a 2D bi-clamped plate domain shown in Figure 3.4. This is a common benchmark domain as seen by the results derived in Section 3.4.1 and 3.4.2. For this problem, the plate is taken to be 1 cm thick, clamped at both ends and L is 0.04 m. The load P is modeled as a traction with magnitude 250 MPa, along a 5 mm edge which has its two endpoints equidistant from the midpoint of the lower

Table 4.2: Candidate isotropic material properties.

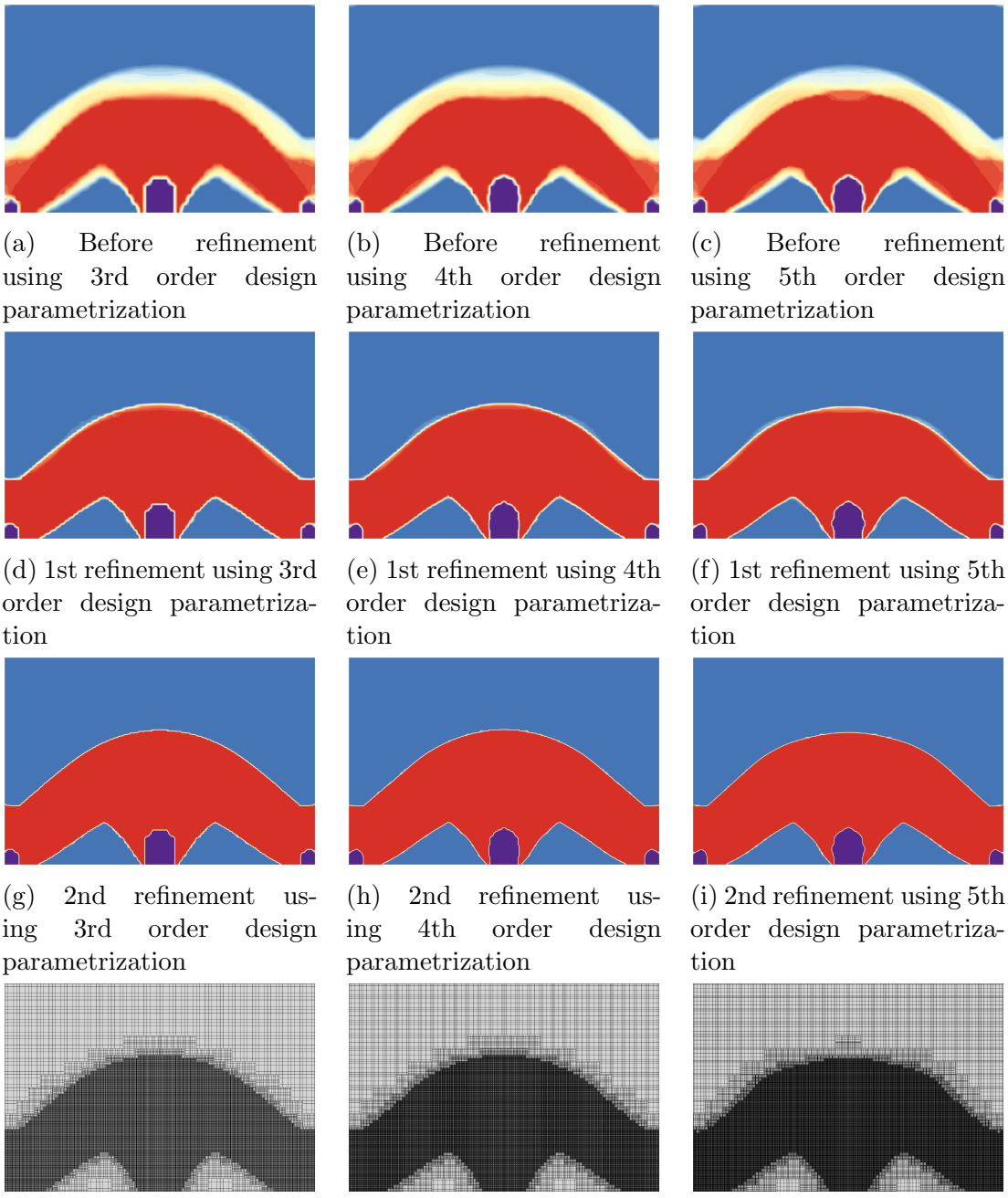
Material Properties	M1	M2
Young's Modulus E (GPa)	70	35
Density ρ (kg/m^3)	2600	1300
Poisson's ratio ν	0.3	0.3
Coefficient of Thermal Expansion α (10^{-6}/°C)	23.5	11.75
Thermal Conductivity (W/(m°C))	130.0	65.0
GTSC β (MPa/°C)	4.113	1.028

edge. The uniform change in temperature field θ is 50°C. The mass is fixed at $0.3\rho_{\text{average}}V$, where V is the volume of the domain and ρ_{average} is the average density of the candidate materials used.

The domain is discretized with an initial mesh size of 52×32 elements with a total mesh size of 1664 elements. Three different mesh orders are used for this study, namely 4th order analysis with 3rd order design mesh, 5th order analysis with 4th order design mesh, and 6th order analysis with 5th order design mesh. These corresponds to multigrid levels of 2, 3, and 4 respectively. 2 cycles of adaptive mesh refinement is used for all mesh cases. The designs are adaptively refined at the 50th and 75th iteration. The initial mesh for the 4th order analysis with 3rd order design has 15229 nodes and 20475 design variables. After the 1st refinement step, the mesh contains 3812 elements, with 34309 nodes and 45747 design variables. After the 2nd refinement step, the final mesh contains 12566 elements, with 112825 nodes and 150255 design variables. The initial mesh for the 5th order analysis with 4th order design has 26961 nodes and 45687 design variables. After the 1st refinement step, the mesh contains 3788 elements with 60601 nodes and 102261 design variables. After the 2nd refinement step, the final mesh contains 12560 elements with 199729 nodes and 336837 design variables. The initial mesh for the 6th order analysis with 5th order design has 42021 nodes and 80883 design variables. After the 1st refinement step, the mesh contains 3788 elements with 94684 nodes and 181791 design variables. After

the 2nd refinement step, the final mesh contains 12482 elements with 311581 nodes and 598011 design variables. The final design for each case is shown in Figure 4.21. Figure 4.21 shows the resulting designs for the three cases with M1 denoted in purple, M2 denoted in red and void in blue. For all three cases, the optimizer produced symmetric designs similar to what was shown in [25] and [84] even though different candidate materials are used. The stiffer M1 is placed along the load application point to minimize the displacement due to it. On the other hand, M2, which has a lower coefficient of thermal expansion α than M1, is chosen by the optimizer to form an arc which counteracts the vertical displacement due to the thermal loads. One slight difference in all three cases is the distribution of M1 around the load application region. Figures 4.21j, 4.21k, 4.21l depict the final analysis meshes that demonstrates the automatic higher order quadrilateral mesh generation for adaptive mesh refinement.

Figure 4.22 shows the convergence of the three cases as a function of iteration history. The compliance value shown is normalized by the initial compliance of the 3rd order design. For the 3rd and 4th order design, the problem is solved using 4 processors, requiring 125 optimization iterations over 12.2 and 31.1 minutes respectively. For the 5th order design, the problem is solved using 6 processors, requiring 125 optimization iterations over 71.0 minutes. To compare the computational effort for each case, the CPU days, which is the product of the number of processors with the optimization wall time, is used as the benchmark. The convergence history of the compliance of these three cases are plotted as a function of its CPU hours as shown in Figure 4.22c. A summary of the optimization outcome for the three cases are in Table 4.3. From Figure 4.22, there are only slight differences in the final compliance objectives for the 3 different mesh cases. For all mesh cases, the final design is obtained after 125 total design iterations, where the change in relative change in the compliance objective γ is less than 10^{-4}. From Table 4.3, the compliance of the 3rd



(a) Before refinement using 3rd order design parametrization

(b) Before refinement using 4th order design parametrization

(c) Before refinement using 5th order design parametrization

(d) 1st refinement using 3rd order design parametrization

(e) 1st refinement using 4th order design parametrization

(f) 1st refinement using 5th order design parametrization

(g) 2nd refinement using 3rd order design parametrization

(h) 2nd refinement using 4th order design parametrization

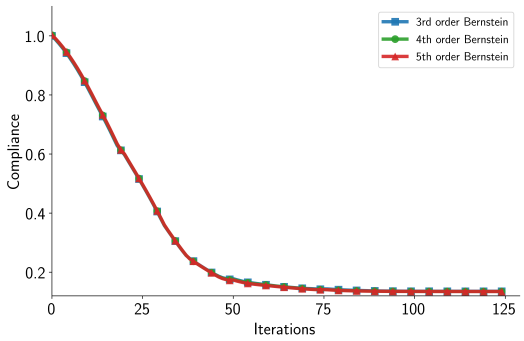
(i) 2nd refinement using 5th order design parametrization

(j) Final analysis mesh using 3rd order design parametrization

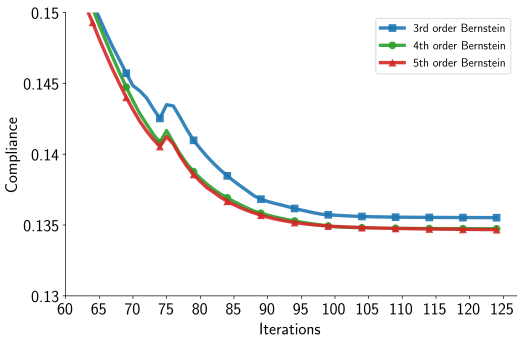
(k) Final analysis mesh using 4th order design parametrization

(l) Final analysis mesh using 5th order design parametrization

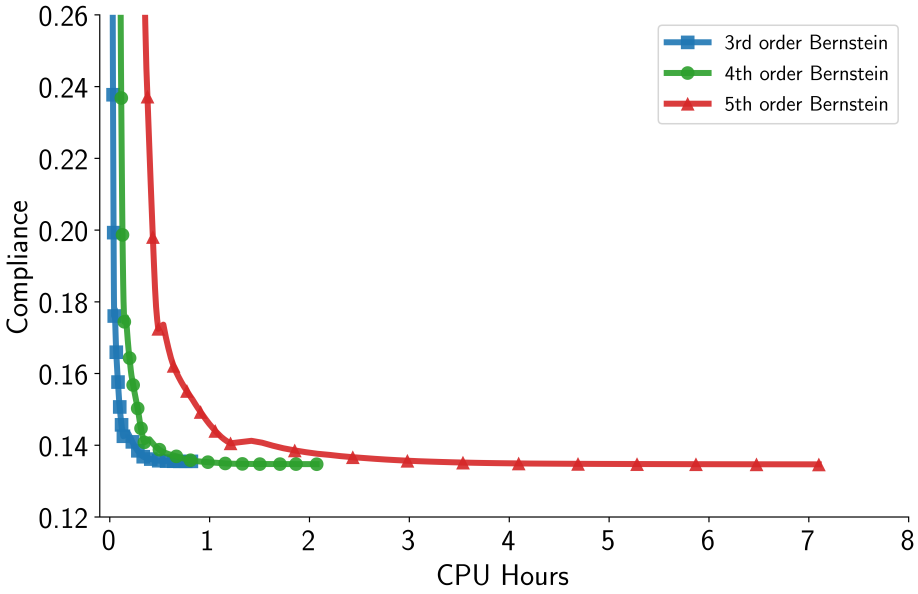
Figure 4.21: Final design of the bi-clamped domain from the three cases with M1 denoted in purple and M2 denoted in red.



(a) Compliance convergence history for the 2D bi-clamped cases.



(b) Zoom view for the compliance convergence history for the 2D bi-clamped cases.



(c) Compliance convergence for the 2D bi-clamped cases as a function of CPU hours

Figure 4.22: Convergence history of the 2D bi-clamped design for the three mesh cases.

Table 4.3: Summary of optimization outcomes for each 2D bi-clamped design case.

Optimization Outcome	3rd order	4th order	5th order
Normalized Compliance	0.1355	0.1347	0.1346
Nodes in initial mesh	15229	26961	42021
Nodes after first refinement	34309	60601	94684
Nodes in final mesh	112825	199729	311581
CPU Cores	4	4	6
Walltime (mins)	12.2	31.1	71.0
CPU hours	0.816	2.07	7.10

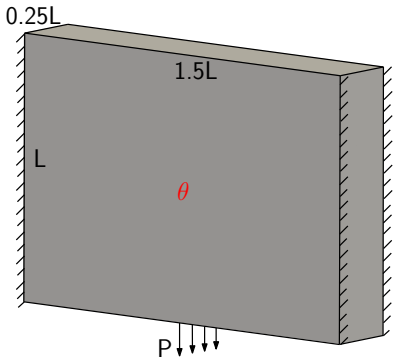


Figure 4.23: Dimensions for the 3D Bi-clamped problem.

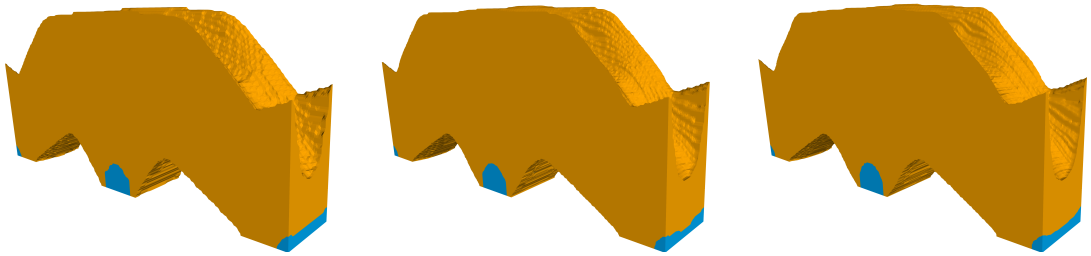
order design parametrization is evaluated to be 0.58% larger compared to that of the 4th order design parametrization and 0.63% larger compared to that of the 5th order design parametrization. The small difference in the compliance for all three mesh cases is not surprising given these designs are similar except in the distribution of M1 along the mechanical load distribution edge. Figure 4.22c illustrates the computational effort required for all three mesh cases, with the combination of the 6th order analysis and 5th order design parametrization taking 7.10 CPU hours, while the 4th order design parametrization took 2.07 CPU hours, signifying a difference of 70.8% in terms of computational resources. Lastly, the 3rd design parametrization took only 0.816 CPU hours, a difference of 88.5% in terms of computational resources compared to the 5th order design parametrization case. For all three designs, the mass constraint is satisfied to machine precision.

The second problem investigated is shown in Figure 4.23. The geometry is a 3D bi-clamped domain, which is the extruded version of the 2D domain shown in Figure 3.4. While the 2D results were obtained using plane stress elements, the 3D results will be obtained using solid elements. For this problem, the plate is clamped at both ends and L is 0.04 m. The load P is modeled as a traction with magnitude 250 MPa on the lower face. The face is the area occupied by a 5 mm edge which has its two endpoints equidistant from the midpoint of the lower edge and through the

thickness 0.25L. The uniform change in temperature field θ is 50°C. The mass is fixed at $0.3\rho_{\text{average}}V$, where V is the volume of the domain and ρ_{average} is the average density of the candidate materials used.

The domain is discretized with an initial mesh size of $52 \times 32 \times 8$ elements with a total mesh size of 13312 elements. Three different mesh orders are used for this study, namely 4th order analysis with 3rd order design mesh, 5th order analysis with 4th order design mesh, and 6th order analysis with 5th order design mesh. 1 cycle of adaptive mesh refinement is used for all mesh cases. The designs are adaptively refined at the 50th iteration.

The initial mesh for the 4th order analysis with 3rd order design has 380725 nodes and 348075 design variables. After the 1st refinement step, the mesh contains 59484 elements, with 1630615 nodes and 1460283 design variables. The initial mesh for the 5th order analysis with 4th order design has 889713 nodes and 1142175 design variables. After the 1st refinement step, the mesh contains 58728 elements with 3800745 nodes and 4828035 design variables. The initial mesh for the 6th order analysis with 5th order design has 1722861 nodes and 2669139 design variables. After the 1st refinement step, the final mesh contains 58448 elements with 7370841 nodes and 11346435 design variables. As mentioned in Section 4.2.1, an under-relaxation coefficient of 0.5 associated with the block Gauss-Seidel smoothing operation is used to improve the convergence of the multigrid-preconditioned GMRES method. The final design for each case is shown in Figure 4.24. Figure 4.24 shows the resulting designs for the three cases with M1 denoted in purple and M2 denoted in red. For all three cases, the optimizer produced symmetric designs that is similar but not identical to the 2D domains, with the stiffer M1 is placed along the load application point to minimize the displacement due to it. M2, which has a lower coefficient of thermal expansion α than M1, is chosen by the optimizer to form an in-plane arc which counteracts the in-plane displacement due to the thermal loads. Due to the thermal



(a) Isometric view of design using 3rd order design parametrization

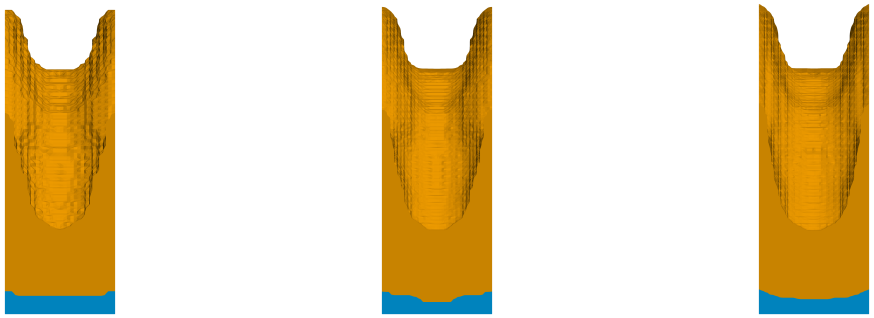
(b) Isometric view of design using 4th order design parametrization

(c) Isometric view design using 5th order design parametrization

(d) Bottom of design using 3rd order design parametrization

(e) Bottom of design using 4th order design parametrization

(f) Bottom of design using 5th order design parametrization



(g) Side of design using 3rd order design parametrization

(h) Side of design using 4th order design parametrization

(i) Side of design using 5th order design parametrization

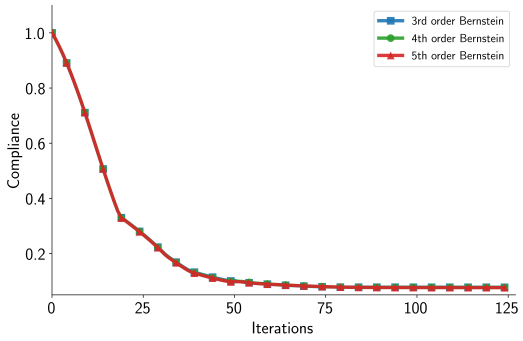
Figure 4.24: Final design of the bi-clamped domain from the three cases with M1 denoted in blue and M2 denoted in yellow.

Table 4.4: Summary of optimization outcomes for each 3D bi-clamped design case.

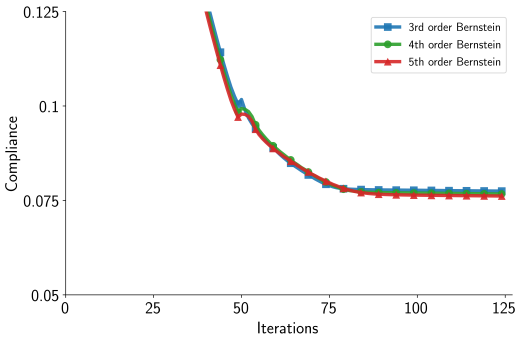
Optimization Outcome	3rd order	4th order	5th order
Normalized Compliance	0.07746	0.07669	0.07615
Nodes in initial mesh	380725	889713	1722861
Nodes in final mesh	1630615	3800745	7370841
CPU Cores	192	192	384
Walltime (hours)	46.1	53.8	92.7
CPU Days	368.9	430.1	1483.5

load causing out-of-plane expansion and the more favorable thermal properties of M2, the through-thickness topology is not identical, with variation in the topology mostly due to distribution of M2.

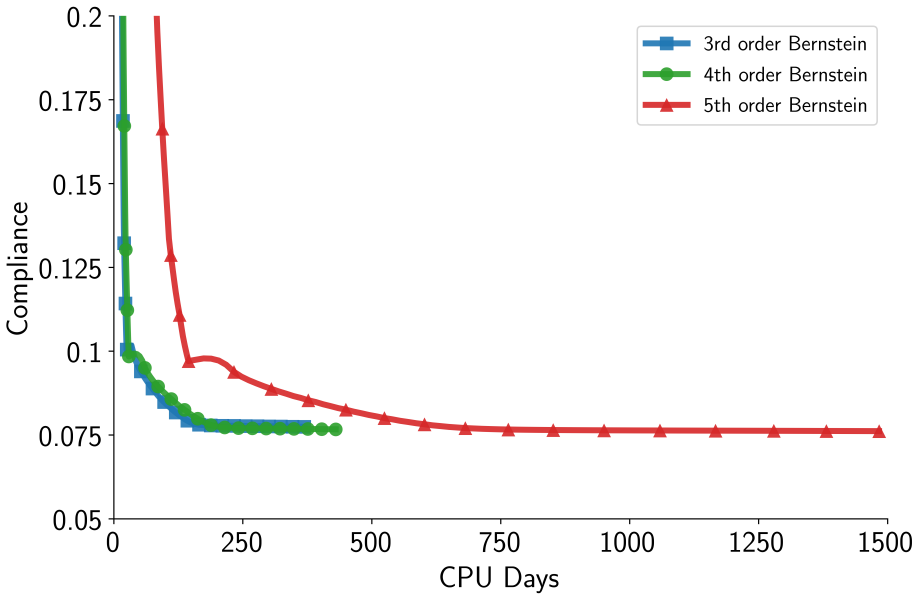
Figure 4.25 shows the convergence of the three cases as a function of iteration history. The compliance value shown is normalized by the initial compliance of the 3rd order design. For the 3rd and 4th order design, the problem is solved using 192 processors, requiring 125 optimization iterations over 46.1 and 53.8 hours respectively. For the 5th order design, the problem is solved using 384 processors, requiring 125 optimization iterations over 92.7 hours. To compare the computational effort for each case, the CPU days, which is the product of the number of processors with the optimization wall time, is used as the benchmark. The convergence history of the compliance of these three cases are plotted as a function of its CPU days as shown in Figure 4.25c. A summary of the optimization outcome for the three cases are in Table 4.4. From Figure 4.25, there are only slight differences to the final compliance objectives for the 3 different mesh cases. For all mesh cases, the final design is obtained after 125 total design iterations, where the change in relative change in the compliance objective γ is less than 10^{-4}. From Table 4.4, the compliance of the 3rd order design parametrization is evaluated to be 0.99% larger compared to that of the 4th order design parametrization and 1.68% larger compared to that of the 5th order design parametrization. The small difference in the compliance for all three mesh cases is not



(a) Compliance convergence history for the 3D bi-clamped cases.



(b) Zoom view for the compliance convergence history for the 3D bi-clamped cases.

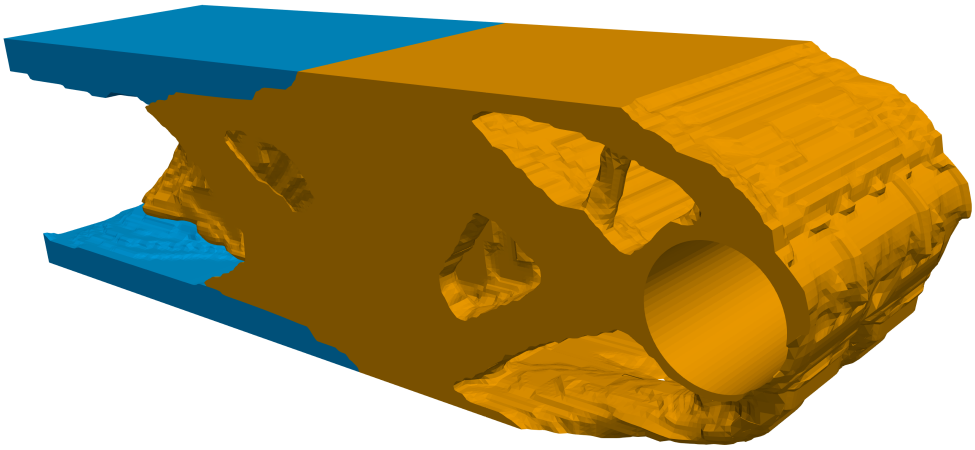


(c) Compliance convergence for the 3D bi-clamped cases as a function of CPU hours

Figure 4.25: Convergence history of the 3D bi-clamped design for the three mesh cases.

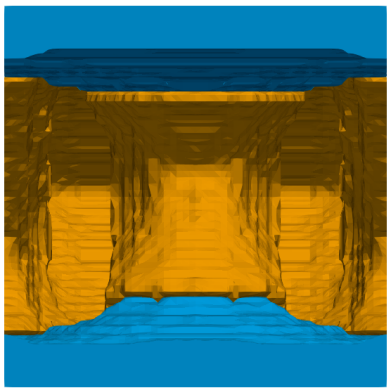
surprising given these designs are similar except in the distribution of M1 along the mechanical load distribution edge. Figure 4.25c illustrates the computational effort required for all three mesh cases, with the combination of the 6th order analysis and 5th order design parametrization taking 1483.5 CPU days, while the 4th order design parametrization took 430.1 CPU days, signifying a difference of 71.0% in terms of computational resources. Lastly, the 3rd design parametrization took only 368.9 CPU days, a difference of 75.1% in terms of computational resources compared to the 5th order design parametrization case. For all three designs, the mass constraint is satisfied to machine precision.

The next example is is a cantilever beam of overall length L shown in Figure 3.16. A hole with diameter 0.1L is cut out at 0.85L from the root of the beam. Traction is applied in the vertical direction throughout the hole. In this work, the value of L is 1 m and the magnitude of the applied traction is 5.8 MPa. The uniform change in temperature field θ is 15°C. The mass is fixed at $0.4\rho_{\text{average}}V$, where V is the volume of the domain and ρ_{average} is the average density of the candidate materials used. For this problem, an external planar mesh, generated in NX [136], is imported into TMR which is used to generate the hexahedral mesh. 1 cycle of adaptive mesh refinement is used with 5 levels of multigrid. The initial mesh used in this study is a 5th order analysis and 4th order design mesh since it gives us a comparable design to the one with 5th order design mesh but in considerably less time. The initial analysis mesh has 4340 elements, 294216 nodes, and 379440 design variables. After refinement at the 150th iteration, the final mesh has 22512 elements with 1448904 nodes and 1837134 design variables. Similar to the previous 3D bi-clamped example, an under-relaxation coefficient of 0.5 associated with the block Gauss-Seidel smoothing operation is used. The final design is shown in Figure 4.26. Figure 4.26 shows the resulting design with M1 denoted in purple and M2 denoted in red. Unlike a purely mechanically loaded case, the resulting design with thermal



(a) Isometric view of design

(b) Front view of design

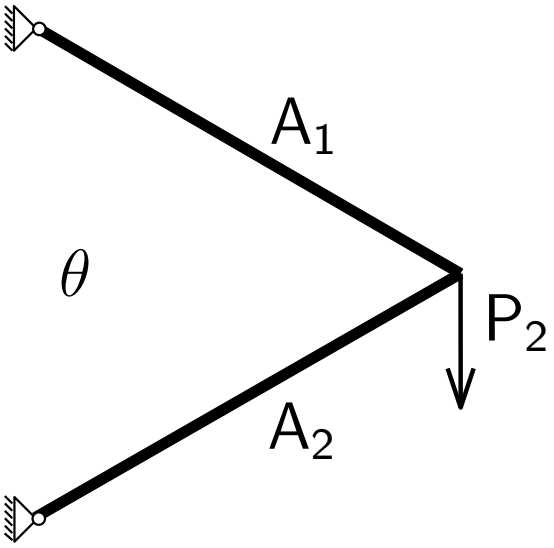


(c) Root view of design

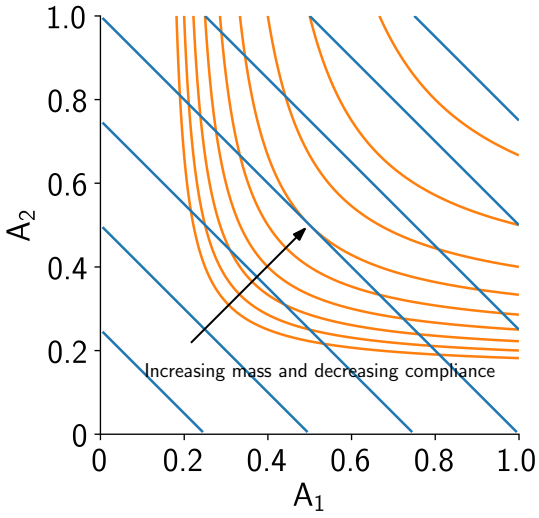
Figure 4.26: Final design of the beam with M1 denoted in blue and M2 denoted in yellow.

loading is not symmetrical, with the upper half of the design having more material i.e. stiffer. This is not surprising given that the combined mechanical and thermal loading is nonsymmetrical. The resulting design can be in part explained by a 2D truss problem as shown in Figure 4.27. Figure 4.27a shows a 2-bar truss problem with combined mechanical and thermal loading. This simplifies the geometry of the 3D cantilever beam problem. Figures 4.27b and 4.27c show the design space with only mechanical loading and combined mechanical and thermal loading respectively. The blue contour lines indicate the mass constraint while the orange contour lines indicate the compliance contour. The design space with only mechanical loading is symmetric which contributes to a symmetric optimized design. On the other hand, from Figure 4.27c, the mass constraint is symmetric but the compliance contour is non-symmetric and it is bias towards A_1, indicating that an optimal design will have a larger A_1, i.e. stiffer structure, compared to A_2. This informs us that the 3D design should be stiffer at the upper half compared to the lower half. Due to the multimaterial nature of the problem, it also make senses that the stiffer upper half is occupied mostly by M2, which is favored by the optimizer due to its lower coefficient of thermal expansion. Figure 4.28 shows the convergence of the optimization and the overall design history with the compliance value normalized by its initial value. The optimization took 49.2 hours on 192 processors over 450 iterations, where the change in relative change in the compliance objective γ is less than 10^{-5} and has a normalized value of 0.164. I observed that the initial compliance quickly converged within the first 50 iterations, with the mass constraint satisfied soon after. At the 150th iteration, the design is not fully discrete. Thus, the design is adaptively refined, and the optimization is continued on this new mesh. This resulted in the spike in the infeasibility of the optimization due to the mass constraint. However, the optimizer drives the mass constraint to machine precision within the next 10 iterations.

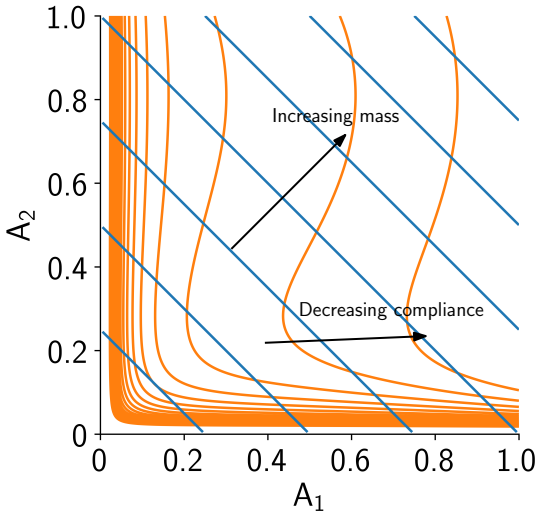
For the last example, a coupled thermoelastic and heat transfer problem is in-



(a) 2D truss design problem



(b) Design space with only mechanical load



(c) Design space with combined mechanical and thermal load

Figure 4.27: Design space of a 2-bar truss problem.

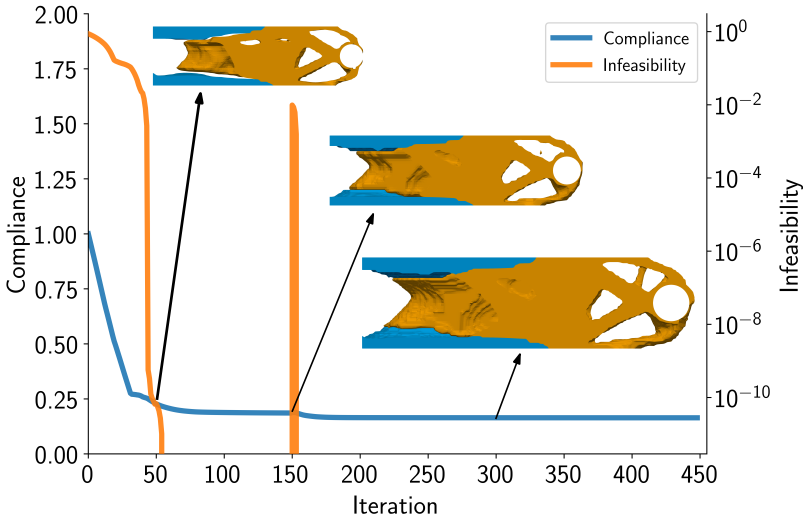


Figure 4.28: Convergence history of the beam design

vestigated. The problem solved in this example is a mass-minimization, stress- and thermal-constrained single-material design. It is formulated as follows:

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & m(\mathbf{x}) \\
\text{such that} \quad & \mathbf{x} \geq 0 \\
& \mathbf{A}_w \mathbf{x} = \mathbf{e} \\
& c_{\mathrm{KS}}(\mathbf{x}, \mathbf{u}; \rho_{\mathrm{KS}}) \leq c_u \\
& c_{\mathrm{KS}}(\mathbf{x}, \boldsymbol{\theta}; \rho_{\mathrm{KS}}) \leq c_\theta \\
\text{governed by} \quad & \begin{bmatrix} \mathbf{K}(\mathbf{x}) & \mathbf{L}(\mathbf{x}) \\ 0 & \mathbf{H}(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{q} \end{bmatrix},
\end{aligned}
\tag{4.20}
$$

where $m(\mathbf{x})$ is the mass of the structure, $\mathbf{u}$ are the displacement state variables, and $\mathbf{x}$ are the design variables. The stress constraint $c_{\mathrm{KS}}(\mathbf{x}, \mathbf{u}; \rho_{\mathrm{KS}})$ is as described in the previous section and c_u is the upper bound on the ratio of the KS value and the failure stress. The thermal constraint $c_{\mathrm{KS}}(\mathbf{x}, \boldsymbol{\theta}; \rho_{\mathrm{KS}})$ is an aggregation of θ in the domain and c_θ is the upper bound on the ratio of the KS value and the maximum allowable θ. The problem domain is shown in Figure 4.29. A hole with diameter 0.1L is cut out

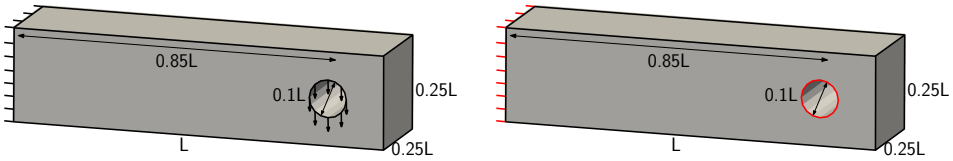


Figure 4.29: Combined thermoelastic and heat transfer cantilever beam problem.

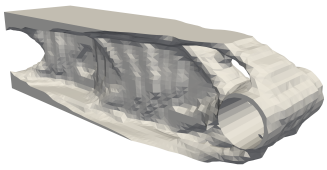
at 0.85L from the root of the beam. For the structural problem, a traction is applied in the vertical direction throughout the hole and it is cantilevered at the root. In this work, the value of L is 1 m and the magnitude of the applied traction is 5.8 MPa. For the heat transfer problem, an inward heat flux is applied throughout the hole and θ is specified at the root. In this work, the magnitude of the applied heat flux is 1300 Wm^{-2} and the specified θ at the root is 0 °C. The mesh used in this study is a 4th order analysis and 3rd order design mesh, with 4340 elements, 126480 nodes, and 38892 design variables. The designs for three different maximum θ constraint is shown in Figure 4.30. Figure 4.30 shows the three designs of the beam with stress and different thermal constraint. It is observed that the beam designs in general follows an I-beam like structure. However, the designs form from the root to the load application region, it becomes more non-symmetric as thermal loads become more significant. Figure 4.31 shows a Pareto front of the different mass values of the final designs with different maximum temperature constraint. From Figure 4.31, I note that in general, the final mass of the design decreases as the maximum allowable θ increases. a reason for this increase is that the additional mass is added near the heat flux location to reduce the maximum θ allowed in the domain. It is also observed that the design with maximum θ of 8.5°C has the lowest mass. This is most likely due to the increase in thermal loading which necessitates the increase in mass required for load bearing.

Through this work, higher order analysis and design parametrization techniques are demonstrated. In particular, the use of Bernstein polynomials are applied to the geometric multigrid solution method used in the framework to allow for its appli-

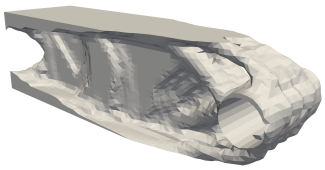
(a) Root view of design with maximum θ of 8.5°C

(b) Root view of design with maximum θ of 9°C

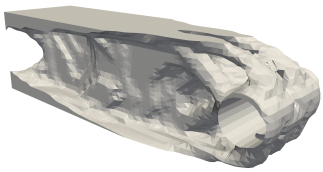
(c) Root view of design with maximum θ of 10°C



(d) Isometric view of design with maximum θ of 8.5°C



(e) Isometric view of design with maximum θ of 9°C



(f) Isometric view of design with maximum θ of 10°C

(g) Front of design with maximum θ of 8.5°C

(h) Front of design with maximum θ of 9°C

(i) Front of design with maximum θ of 10°C

Figure 4.30: Final design of the domain from the three different maximum θ constraint.

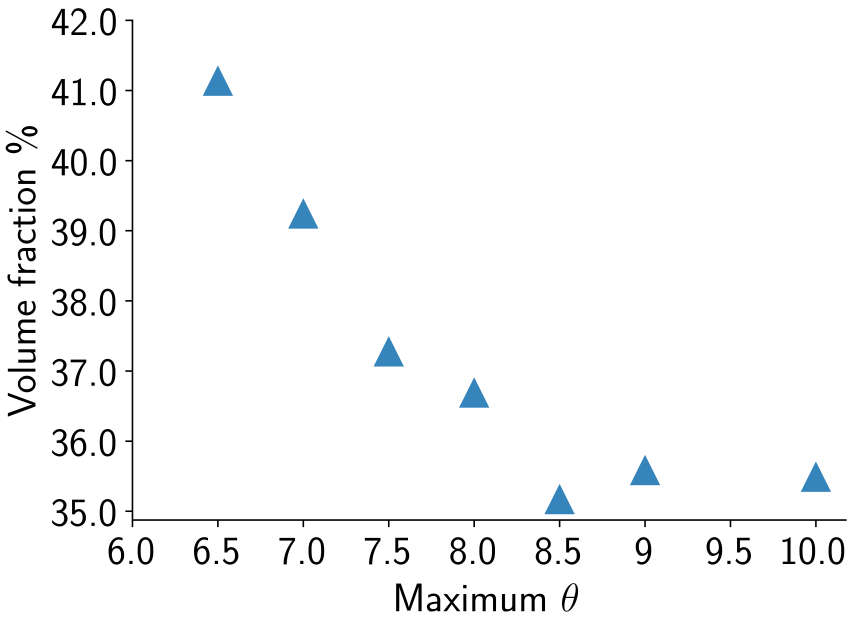


Figure 4.31: Final design volume fraction as a function of maximum θ

cation to large-scale design optimization. Using the higher order Bernstein polynomials allows for the optimizer to vary the topology/material within each element. Its application to multimaterial thermoelastic topology optimization problems were illustrated.

CHAPTER 5
TOPOLOGY OPTIMIZATION WITH NATURAL FREQUENCY CONSTRAINTS

In this chapter, efficient techniques for evaluation of natural frequency constraint of a large-scale design optimization are developed. Natural frequency constraints are added to design problems when it is subjected to high vibration environments. Such environment may also present other design consideration such as thermal loading in the instance of hypersonic reentry vehicles. Literature presented different techniques used for natural frequency-constrained topology optimization but the most common method is Lanczos which is computationally expensive to employ for large-scale designs. This means that designs involving Lanczos method have limited design freedom and the meshes are much coarser, thus restricting the design space and mesh resolution. By employing a suitable large-scale eigensolver with eigenvector recycling strategy, large-scale designs become amenable and designs with finer mesh resolution can be obtained.

The Natural Frequency Eigenvalue Problem

Finding the natural frequencies of vibration requires the solution of the generalized eigenvalue problem

$$\mathbf{K}(\mathbf{x})\mathbf{u}_i = \lambda_i \mathbf{M}(\mathbf{x})\mathbf{u}_i, \tag{5.1}$$

where λ_i is the ith eigenvalue and $\mathbf{u}_i$ is the corresponding eigenvector.

The purpose of the natural frequency constraint is to apply a lower bound to the fundamental frequency. An obstacle with applying this bound is the mathematical issue of mode switching, which makes the eigenvalues locally non-differentiable [96],

and therefore incompatible with gradient-based optimization. Mode switching occurs when the eigenmode associated with the lowest eigenvalue switches as the design evolves. This is expected for topology optimization where drastic changes to the initial provided design is not uncommon. At the cross-over point, the minimum eigenvalue is not differentiable [96]. Authors have addressed this issue using the bound formulation with modal assurance techniques [99, 100], or using p-norm or KS aggregation strategies [145]. In this work, the r-lowest eigenvalues, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_r$, are aggregated through KS aggregation, resulting in the following constraint

$$c_{KS_\lambda}(\lambda_1, \ldots, \lambda_r) = \lambda_1 - \frac{1}{\rho_{KS_\lambda}} \ln \left[\sum_{i=1}^{r} e^{-\rho_{KS_\lambda}(\lambda_i - \lambda_1)} \right] \geq \omega_0^2, \tag{5.2}$$

where $\rho_{KS_\lambda} = 50$. In this way, if mode switching occurs, the KS-aggregation still identifies the approximate lowest eigenvalue, rendering eigenmode tracking as redundant.

Evaluating the eigenvalue aggregate (5.2) requires the solution of a large-scale generalized eigenvalue problem. Many eigenvalue solution methods utilize direct factorization techniques, making them too computationally expensive for large-scale topology optimization problems. In this work, Jacobi–Davidson method [41] is developed to compute the eigenvalues of the natural frequency problem (5.1). This method leverages the scalable geometric multigrid preconditioner used in the Krylov solution method for the finite-element governing equations. To accelerate the eigenvalue solution procedure, I propose two eigenvector recycling strategies, which utilize eigenvectors from the eigenproblem at the previous design iteration to provide an initial subspace. Eigenvector recycling has been used in the context of nonlinear eigenvalue problems [146], but have not been investigated in the context of topology optimization. Recycling methods for the solution of linear systems have been demonstrated for topology optimization in the context of mass-constrained compliance minimiza-

tion [48, 147], but not, to the best of our knowledge, for eigenvalue problems. To compare the performance of the proposed method, the eigenproblem (5.1) is also solved using a conventional shift-and-invert Lanczos method [44].

Shift-and-invert Lanczos method

Shift-and-invert Lanczos techniques are commonly used in finite-element frequency and buckling analysis [44]. The shift-and-invert strategy preconditions the spectral properties of the eigenproblem (5.1) to promote separation of the eigenvalues close to a desired value, thereby accelerating the convergence of the Lanczos method. With a shift-and-invert strategy, the natural frequency eigenproblem (5.1) becomes

$$\mathbf{M}(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{u}_i = \mu_i \mathbf{M}\mathbf{u}_i, \tag{5.3}$$

where the transformed eigenvalues are $\mu_i = 1/(\lambda_i - \sigma)$ and the original eigenvalues that are close to the shift value σ become the extreme eigenvalues of the transformed eigenproblem. Shift-and-invert Lanczos methods have proven to be very effective when a full factorization of the matrix $\mathbf{K} - \sigma\mathbf{M}$ is available. While Lanczos-type methods have be proven to be robust, a full factorization is not computationally feasbile for large-scale applications. Instead, iterative solution methods are required. Unfortunately, shift-and-invert strategies require a tightly-converged solution for every application of the operator $(\mathbf{K} - \sigma\mathbf{M})^{-1}$, making them expensive when combined with iterative methods. In contrast, the Jacobi–Davidson method can be used with inexact solutions of a linear system without sacrificing the accuracy of the method.

Shift-and-invert Lanczos methods can be used for generalized eigenproblems by working with the problem (5.3), combined with a $\mathbf{M}$-orthogonal subspace. The Lanczos method constructs an approximation that takes the form $\mathbf{u} = \mathbf{V}_k\mathbf{y}$, where $\mathbf{V}_k \in \mathbb{R}^{n\times k}$. $\mathbf{M}$-orthonormalization ensures that $\mathbf{V}_k^T\mathbf{M}\mathbf{V}_k = \mathbf{I}$. As a result, the

approximation can be formed as follows

$$\mathbf{V}_k^T\mathbf{M}(\mathbf{K}-\sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{V}_k^T\mathbf{y} \triangleq \mathbf{T}_k\mathbf{y} = \mu\mathbf{V}_k^T\mathbf{M}\mathbf{V}_k\mathbf{y} = \mu\mathbf{I}$$

Here $\mathbf{T}_k$ is a symmetric tridiagonal matrix formed from the Lanczos process.

Jacobi–Davidson method

The goal of the Jacobi–Davidson method is to find approximate solutions of the generalized eigenproblem (5.1) through an iterative technique that uses the Davidson approach of constructing an approximation to the eigenvector using a **M**-orthogonal subspace, while using Jacobi's method to search for new vectors to add to this subspace [41]. A detailed description of the algorithm is shown in Algorithm 4.

At iteration k of the Jacobi–Davidson method, an **M**-orthogonal subspace, $\mathbf{V}_k \in \mathbb{R}^{n\times k}$, is formed and satisfies the property

$$\mathbf{V}_k^T\mathbf{M}\mathbf{V}_k = \mathbf{I}. \tag{5.4}$$

The approximate eigenvalues and eigenvectors, called the Ritz values and Ritz vectors, are denoted as $\lambda_i \approx \theta_i$, and $\mathbf{u}_i \approx \mathbf{V}_k\mathbf{y}_i$. The Ritz values and vectors are obtained by enforcing a Galerkin orthgonality condition

$$\mathbf{K}\mathbf{V}_k\mathbf{y}_k - \theta\mathbf{M}\mathbf{V}_k \perp \mathbf{V}_k,$$

which leads to the reduced $k \times k$ eigenproblem

$$\mathbf{V}_k^T\mathbf{K}\mathbf{V}_k\mathbf{y}_i = \theta_i\mathbf{V}_k^T\mathbf{M}\mathbf{V}_k\mathbf{y}_i.$$

Defining the matrix $\mathbf{A}_k \triangleq \mathbf{V}_k^T\mathbf{K}\mathbf{V}_k \in \mathbb{R}^{k\times k}$, and using the **M**-orthogonality prop-

Algorithm 4 Jacobi–Davidson method with recycling

Set $k = 0$
while $k \leq s$ **do** ▷ Orthogonalize initial set of recycled vectors
  Set $\mathbf{v}_k \leftarrow \text{MGS}(\mathbf{M}, \mathbf{V}_{k-1}, \mathbf{R}_s\mathbf{e}_k)$ ▷ Extract column k from $\mathbf{R}_s$ and orthogonalize it
  Set $k = k + 1$
end while
Compute $\mathbf{A}_s = \mathbf{V}_s^T\mathbf{K}\mathbf{V}_s$ ▷ Compute the initial reduced matrix
Set $\mathbf{E}_k = \emptyset$ ▷ Set the converged eigenvectors to the empty set
Set $k = s$
while $k \leq$ max iterations **do**
  if $k > s$ **then** ▷ Skip this orthogonalization step when $k = s$
    Set $\mathbf{v}_k \leftarrow \text{MGS}(\mathbf{M}, \mathbf{E}_k, \mathbf{v}_k)$ ▷ Orthogonalize $\mathbf{v}_k$ against converged eigenvectors $\mathbf{E}_k$
    Set $\mathbf{v}_k \leftarrow \text{MGS}(\mathbf{M}, \mathbf{V}_{k-1}, \mathbf{v}_k)$ ▷ Orthogonalize $\mathbf{v}_k$ against $\mathbf{V}_{k-1}$
    Compute $\mathbf{w} = \mathbf{K}\mathbf{v}_k$
    Compute new row/column of $\mathbf{A}_k$ where $[\mathbf{A}_k]_{jk} = \mathbf{w}^T\mathbf{v}_j$ and $[\mathbf{A}_k]_{kj} = \mathbf{w}^T\mathbf{v}_j$
  end if
  Solve the eigenproblem $\mathbf{A}_k\mathbf{y} = \theta\mathbf{y}$
  Compute for the lowest Ritz vector $\mathbf{u}_1 = \mathbf{V}_k\mathbf{y}_1$
  Compute the residual $\mathbf{r} = \mathbf{K}\mathbf{u}_1 - \theta_1\mathbf{M}\mathbf{u}_1$
  if $||\mathbf{r}||_2 \leq \epsilon||\mathbf{K}\mathbf{u}_1||_2$ **then** ▷ Check for convergence of this Ritz pair
    Add $\mathbf{u}_1$ to the converged eigenvectors $\mathbf{E}_k = \mathbf{E}_{k-1} \cup \mathbf{u}_1$
    if required eigenvectors converged **then**
      break ▷ All eigenvalues and eigenvectors converged
    end if
    Compute $\mathbf{u}_2 = \mathbf{V}_k\mathbf{y}_2$, $\mathbf{r} = \mathbf{K}\mathbf{u}_2 - \theta_2\mathbf{M}\mathbf{u}_2$ ▷ Switch to the next Ritz pair
    Set $\mathbf{u}_1 \leftarrow \mathbf{u}_2$
  else
    Set $\mathbf{E}_k = \mathbf{E}_{k-1}$
  end if
  Set $\mathbf{Q}_k = \mathbf{E}_k \cup \mathbf{u}_1$
  Use FGMRES to approximately solve the update equation

$$(\mathbf{I} - \mathbf{M}\mathbf{Q}_k\mathbf{Q}_k^T)(\mathbf{K} - \theta\mathbf{M})(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^T\mathbf{M})\mathbf{t} = -\mathbf{r}$$

  Set $\mathbf{v}_{k+1} = \mathbf{t}$
  Set $k = k + 1$
end while

erty (5.4)

$$\mathbf{A}_k \mathbf{y}_i = \theta_i \mathbf{y}_i. \tag{5.5}$$

Since the dimension of the subspace is small, where $k \ll n$, a solution method for small dense eigenproblems can be used to solve (5.5). The Ritz value θ_i and vector $\mathbf{u}_i$ are approximations to the original eigenproblem, so the residual of the Ritz pair $\mathbf{r}_i = (\mathbf{K} - \theta_i \mathbf{M})\mathbf{V}_k \mathbf{y}_i$ is non-zero and an indicator of the accuracy of the approximation.

In this particular implementation of the Jacobi–Davidson method with recycling, for the first s iterations, the subspace $\mathbf{V}_k$ is built using recycled eigenvectors from previous eigenproblems, where s depends on the recycling scheme used. After this initial recycling phase, $\mathbf{V}_k$ is built using vectors generated from an inexact Newton solution. To motivate the update scheme, consider the following Newton-step on the residuals of the i-th generalized eigenvalue appended with the $\mathbf{M}$-normality condition $\mathbf{u}_i^T \mathbf{M} \mathbf{u}_i = 1$

$$\begin{bmatrix} (\mathbf{K} - \theta_i \mathbf{M}) & \mathbf{M}\mathbf{u}_i \\ \mathbf{u}_i^T \mathbf{M} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -(\mathbf{K} - \theta_i \mathbf{M})\mathbf{u}_i \\ \frac{1}{2}(\mathbf{u}_i^T \mathbf{M} \mathbf{u}_i - 1) \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_i \\ 0 \end{bmatrix}, \tag{5.6}$$

which gives the update $(\mathbf{t}, \Delta\lambda)$ to the Ritz pair. Note that by construction the Ritz vector is in the span of the basis, i.e. $\mathbf{u}_i \in \operatorname{span}\{\mathbf{V}_k\}$, and the residual is orthogonal to the basis such that $\mathbf{r}_i^T \mathbf{V}_k = 0$. As a result, the approximate eigenvector $\mathbf{u}_i$ satisfies

$$\mathbf{u}_i^T \mathbf{r}_i = 0.$$

As a consequence of this identity $(\mathbf{I} - \mathbf{M}\mathbf{u}_i \mathbf{u}_i^T)\mathbf{r}_i = \mathbf{r}_i$. The second condition in the linear system (5.6) imposes $\mathbf{u}_i^T \mathbf{M} \mathbf{t} = 0$ such that $\mathbf{t} = (\mathbf{I} - \mathbf{u}_i \mathbf{u}_i^T \mathbf{M})\mathbf{t}$. Combining these two results, the Newton update (5.6) can be written as

$$(\mathbf{I} - \mathbf{M}\mathbf{u}_i \mathbf{u}_i^T)(\mathbf{K} - \theta_i \mathbf{M})(\mathbf{I} - \mathbf{u}_i \mathbf{u}_i^T \mathbf{M})\mathbf{t} = -\mathbf{r}_i. \tag{5.7}$$

Instead of solving for the update (5.7) to a tight tolerance, it can be beneficial to use a loose tolerance that requires fewer iterations. In this work, the same geometric multigrid preconditioner as the linear solver is used and the Krylov-subspace method FGMRES(m) is employed to loosely solve (5.7) where $m = 10$.

Recycling methods for Jacobi–Davidson method

Since the proposed Jacobi–Davidson method is used in a design optimization process, it will be repeatedly applied to a sequence of related eigenproblems. In particular, the eigenvectors will exhibit similar characteristics between optimization iterations, and the difference in the eigenvalues between iterations becomes minute as the design converges. To take advantage of this property, the eigenvectors computed at the previous iterations can be recycled to accelerate the convergence of the next eigenproblem. The Jacobi–Davidson method is well suited to a variety of recycling strategies since the subspace vectors, $\mathbf{V}_k$, need only be $\mathbf{M}$-orthogonal. An $\mathbf{M}$-orthogonal set of vectors can easily be obtained from any set of vectors by applying the modified Gram–Schmidt(MGS) algorithm [148].

The recycling technique starts by constructing a set of an initial set of s vectors, stored as columns in $\mathbf{R}_s \in \mathbb{R}^{n\times s}$ that are computed from the eigenvectors obtained in the previous iteration, as shown in Algorithm 4. Two recycling schemes are examined: (1) one recycled vector, $s = 1$, that is an equally-weighted linear combination of the eigenvectors from the previous solution, and (2) a number of recycled vectors, $s \leq r$, that are equal to the lowest eigenvectors from the previous solution.

The first step in the recycling algorithm is to perform modified Gram–Schmidt to re-orthogonalize the set of recycled vectors and store them in the first s-columns of the basis $\mathbf{V}_k \in \mathbb{R}^{n\times k}$, with $k = s$. Next, the algorithm forms the portion of the $\mathbf{A}_k \in \mathbb{R}^{k\times k}$ matrix formed by $\mathbf{A}_k = \mathbf{V}_k^T\mathbf{K}\mathbf{V}_k$. Finally, the regular Jacobi–Davidson method is started from iteration $k = s$.

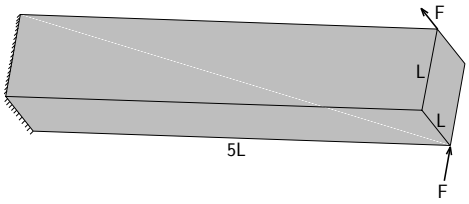


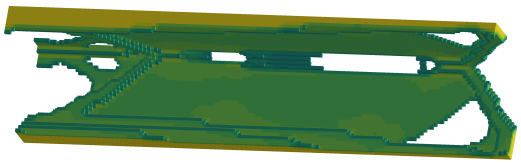
Figure 5.1: 3D cantilever beam with point loads

Performance of Shift-and-Invert Lanczos and Jacobi–Davidson

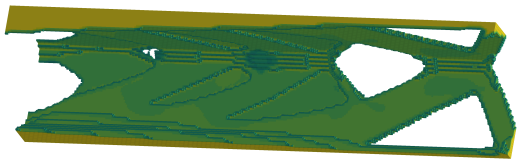
To evaluate the merits of Jacobi–Davidson method over the shift-and-invert Lanczos, a compliance-minimization study with mass and frequency constraints for a 3D beam problem, as shown in Figure 5.1, is performed. The optimization problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \mathbf{u}^{\mathbf{T}}\mathbf{K}(\mathbf{x})\mathbf{u} \\ \text{such that} \quad & 0 \leq \mathbf{x} \leq 1 \\ & c_{KS_\lambda}\left(\lambda_1, \ldots, \lambda_r\right) \geq \omega_0^2 \\ & \mathbf{c}(x) = m_{\text{fixed}} - m(\mathbf{x}) \geq 0 \\ \text{governed by} \quad & \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} \end{aligned} \tag{5.8}$$

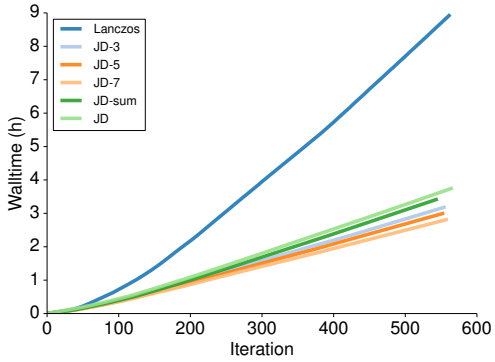
The beam domain is 5L × L × L. In this work, the value of L is 1 m. The mass is fixed at $0.1\rho V$, where V is the volume of the domain and the lower bound on the natural frequency constraint ω_0^2 is set at 2×10^4 Hz^2. The number of lowest eigenvalue aggregated, r, is restricted to $r = 10$. Two different meshes are used for this study: the first mesh has $32 \times 32 \times 160$ elements with $525,987$ DOF and the second mesh has $64 \times 64 \times 320$ elements with over 4 million DOF. The Krylov-subspace method used is the multigrid-preconditioned FGMRES(100) with a maximum of two restarts. The first mesh case is solved using 24 processors and the second mesh case is solved using 48 processors. For each mesh case, the Lanczos method as well as the Jacobi–Davidson method with and without recycling are used. Figure 5.2 shows



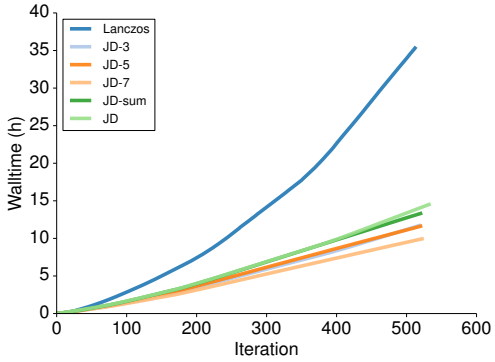
(a) Design obtained using 24 processors for mesh with 525,987 degrees of freedom.



(b) Design obtained using 48 processors for mesh with 4 million degrees of freedom.



(c) Cumulative wall time using 24 processors for mesh with 525,987 degrees of freedom.



(d) Cumulative wall time using 48 processors for mesh with 4 million degrees of freedom.

Figure 5.2: Design and wall time for the different methods on different mesh sizes

a summary of the final structural layout and cumulative computational wall time as a function of iteration history. Figure 5.2c and Figure 5.2d show a comparison between between the total computational time using Lanczos and different variants of the Jacobi–Davidson for the first and second case, respectively. For the first mesh case, the Jacobi–Davidson method, denoted by JD, required between 56 to 70% less computational time than the shift-and-invert Lanczos method. The second mesh case exhibits similar time savings, ranging between 60 and 73% less computational time depending on the recycling strategy used for the Jacobi–Davidson method. Note that even as the design evolves, the different variants of Jacobi–Davidson methods for both mesh sizes falls within the same performance range. As expected, the Jacobi–Davidson method using recycling schemes outperforms the Jacobi–Davidson method that does not use it since the latter discards the eigenvector information from the previous solution. The Jacobi–Davidson method that uses one recycled vector that is an equally-weighted linear combination of the eigenvectors from the previous solution, as denoted by JD-sum, only outperformed the Jacobi–Davidson method without any recycling. Lastly, JD-s, where s is the lowest s eigenvectors recycled from the previous solution, demonstrated better performance as the number of eigenvectors recycled from the previous solution increases.

Through this work, it is shown that the use of Jacobi–Davidson method with the proposed eigenvector recycling strategies can accelerate the solution method used to solve eigenproblems at each topology optimization iteration. Previous works have employed direct method such as the shift-and-invert Lanczos but they are incompatible with large-scale topology optimization problems. The proposed solution method was able to reduce the computational effort by up to 70% with the use of eigenvector recycling on a problem with over $\mathcal{O}(10^6)$ DOF.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

Conclusions

The present work formulated a multimaterial thermoelastic topology optimization framework with capability for local mesh refinement. This modular framework is geared towards solving large-scale design problems with as many as $\mathcal{O}(10^8)$ DOF efficiently and can be extended easily. The framework features geometric multigrid enabled finite element structural solver TACS, optimizer ParOpt and unstructured mesh generation and automated adaptive mesh refinement capability through TMR. TACS can handle large-scale design problems by assembling and solving the linear governing system of equations through geometric multigrid approach that is parallel and scalable. ParOpt is an optimizer that uses the Sℓ_1QP method with an ℓ_∞ trust-region globalization strategy to solve topology optimization problems. It also takes advantage of the sparse nature of the linear constraints arising from DMO multimaterial topology optimization problems. TMR has the ability to automatically generate linear and higher order meshes with local refinement that conforms to the CAD geometry.

To demonstrate the effectiveness of the framework, a scalability study on the algorithms was performed. It was shown that the solution has an overall over 90% efficiency. The framework has also been applied to the single material and multimaterial design with and without thermal loading. In the case without thermal loading, some of the largest 3D multimaterial compliance-minimization designs, with almost $\mathcal{O}(10^9)$ DOF, are presented. With uniform thermal loading, 2D single material and multimaterial problems were investigated for both compliance-minimization and

stress-constrained designs. It was shown that the adaptively refined designs are similar to the designs obtained in existing literature without adaptive mesh refinement. In addition, trade studies for multimaterial stress-constrained problems were carried out to identify possible local minimas. In the case of nonuniform thermal loading, 2D single material compliance-based designs were derived. The framework enables various numerical and design optimization studies. For 3D multimaterial design, a mesh study of various mesh resolution with and without adaptive mesh refinement for a cantilever beam is carried out. I found that for the same final mesh resolution, the difference in the objective for the adaptive mesh refined (fine resolution) design and that for the uniformly fine design (ultra-fine resolution) design is only 1.86% but the ultra-fine resolution design took 36.1% more computational resources. Thus, using adaptive mesh refinement allows for identification of designs that have similar performance to designs derived on a uniform mesh but with less computational resources.

Due to the modular nature of the framework, higher order analysis and design parametrization can be included within the framework without need for major algorithmic changes. The higher order analysis comes in the form of higher order Lagrange polynomials whereas the higher order design parametrization is through Bernstein polynomials, a subset of B-spline polynomials. Combining these parametrization techniques with geometric multigrid, polynomial order up to $p = 6$ for both 2D quadrilateral and 3D hexahedral design problems were solved. Based on the 2D and 3D bi-clamped example presented in the work, it is noted that a 5th order analysis coupled with a 4th order design parametrization results in the best combination of computational resources and objective value when compared to a 6th order analysis coupled with a 5th order design parametrization. For the 2D example, the difference in objective for both parametrization is 0.75% while the 4th order design parametrization has an improvement of 56.1% over the 5th order design. For the 3D example

bi-clamped study, the computational advantages are more overwhelming, with an improvement of 71.0% in terms of computational resources used and less than 1% difference in final objective.

To solve natural frequency-constrained topology optimization problems, an iterative technique known as the Jacobi–Davidson method is used in conjunction with geometric multigrid solution procedures to solve eigenproblems at each design iteration. This technique performs favorably when compared with direct methods such as Lanzcos. Novel eigenvector recycling schemes for Jacobi–Davidson methods were devised and it was shown to be very effective in terms of accelerating the solution procedure, with an improvement in walltime of as much as 73% over Lanczos method.

There are several potential applications based on the contributions from this work. First is the study of combined natural frequency and thermoelastic topology optimization problems such as nacelle or vehicle chassis design. While such designs are not shown in this work, it is straightforward to combine these capabilities in the framework to create these designs. Another potential application arising from this work is the study of non-isotropic thermal properties on coupled heat transfer problem with thermoelastic design. In this design, the material will have more favorable thermal properties, i.e. heat conductivity and coefficient of thermal expansion in certain loading directions. Lastly, a study of the efficiency of higher order design parametrization can be made where a comparison against linear Lagrange design parametrization. With the study of thermal-elastic problems, its potential impacts include more realistic structural design and better design performance based on actual mechanical and thermal loading.

Future Work

Based on the work presented, there are several areas of future research. These generally fall into areas of (1) algorithmic developement and enhancements to topology

optimization, (2) enhacements and extension of physical modeling such as including material and geometrically nonlinear problems, and (3) integration with vehicle design problems. These areas are described in detail below.

Algorithmic enhancements

An area of algorithmic enhancement is to employ the framework for more practical design applications. This means moving away from the canonical problems being solved in this work and moving towards designing more complex geometry. This necessitates making extensions to the mesh generation tool and solution method of the framework. One of the extension to the framework is to enhance the robustness of the automated hexahedral mesh generation algorithm in TMR. Currently, TMR can handle single volume mesh generation robustly. However, for more complex geometry, the domains can be broken into multiple volumes for meshing and in this case, TMR requires more active user input in order to mesh it. Thus, automating this process allows for more practical application. Another enhancement to the framework is in improving the current solution method. As mentioned in Section 2.1.1, a geometric multigrid preconditioner is used to solve the system of equation. On all but the coarsest mesh, a block symmetric Gauss–Seidel smoothing operation is performed while on the coarsest mesh, a parallel direct Schur-complement method is used to solve the reduced linear system. However, the direct method restricts the problem size on the coarsest level and the overall efficiency of the solution method. A more scalable technique would be to use an iterative algebraic multigrid method (AMG) [149] to obtain the solution on the coarsest mesh. AMG can be applied regardless of the geometry and thus suitable for solving linear systems on unstructured coarse meshes.

Another area is to include a higher order filtering scheme that is based on length scale. The current filtering scheme is based on the filtering of the nodal design variables of the design mesh which means that the designs obtained are dependent on

the mesh used. Also, this filtering scheme does not allow for an explicit control over the feature size of the final design. A Helmholtz-type filter [89] was investigated but the unstructured nature of the meshes used for this work rendered the filter unusable. A higher order filtering scheme with length scale control can leverage upon the higher order Bernstein polynomials that are used in this work.

Another area to further explore upon is through the adaptive mesh refinement heuristics. Presently, the adaptive mesh refinement heuristics used is feature-based, where the mesh is adaptively refined at locations where there is material and coarsen where no structure is formed. Other possible adaptive mesh refinement heuristic includes solution-based methods [61] and functional output-based methods [64]. These methods require formulation of an error indicator based on variables such as displacements or stresses and using it as the criterion for adaptive mesh refinement.

Modeling nonlinear physics

An extension to the physics of the current model is to move away from the current steady-state heat transfer thermoelastic problem to a coupled diffusion-elasticity problem formulation. This requires a transient formulation of the heat transfer problem where the temperature field will vary with time. This will result in a fully coupled elasticity and diffusion problem [131] instead of the one-way coupled formulation used in this work.

Another extension will be to incorporate multiscale physics to the analysis so that the optimizer can design both the microstructure i.e. the structural layout for each cell, and the macrostructure i.e. the distribution of said cell in the domain concurrently. This opens a new subset of design space for multimaterial thermoelastic topology optimization and possibilities to further improve on current designs.

Another enhancement to the physics used in the present work is to account for nonlinearity in the geometry of the design domain. There are works on geometric non-

linearity for topology optimization and its effects for both mechanically loaded [150] as well as thermally loaded designs [151]. For significantly larger mechanical and thermal loads, geometric nonlinearity has to be included. For stress-based multi-physics design, geometric nonlinearity can alleviate residual stresses in the structure through local buckling.

Vehicle integration problems

Lastly, an area of future work is integrating with vehicle design problems. This includes overall vehicle design and the design of its subsystem. This pertains to using vehicle performance at operating points as loads and constraints on the topology optimization problem. Different examples include design of a motor nacelle where the loads and thermal constraints on the design are determined by the operating trajectory of the vehicle, and a coupled aerostructural design where the topology optimization determines the distribution of the structural layout based on the prescribed thermal boundary conditions as well as the aerodynamic flow that contributes both mechanical and thermal loading which evolves with the changing topology.

APPENDIX A

APPENDIX

2D THB-spline Constraint Formulation

The following derivation of constraint uses the domain shown in Figure 4.8. In 2D, the spline surface, w, can be expressed as

$$\begin{aligned} w = {} & aa_{0,0}\mathrm{N}_{0x}\mathrm{N}_{0y} + aa_{0,1}\mathrm{N}_{0x}\mathrm{N}'_{1y} + ab_{0,2}\mathrm{N}_{0x}\mathrm{M}_{2y} \\ & + aa_{1,0}\mathrm{N}'_{1x}\mathrm{N}_{0y} + aa_{1,1}\mathrm{N}'_{1x}\mathrm{N}'_{1y} + ab_{1,2}\mathrm{N}'_{1x}\mathrm{M}_{2y} \\ & + ba_{2,0}\mathrm{M}_{2x}\mathrm{N}_{0y} + ba_{2,1}\mathrm{M}_{2x}\mathrm{N}'_{1y} + bb_{2,2}\mathrm{M}_{2x}\mathrm{M}_{2y} \end{aligned} \tag{A.1}$$

For $[2.0, 2.5]\times[2.0, 2.5]$, the active B-spline functions are N_{0x}, N'_{1x} and M_{2x} in the x direction and N_{0y}, N'_{1y} and M_{2y} in the y direction. Making the appropriate substitutions and simplification,

$$\begin{aligned} w = {} & \mathrm{M}_{1x}\mathrm{M}_{1y}\left(\frac{9}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{1}{16}aa_{1,1}\right) \\ & + \mathrm{M}_{1x}\mathrm{M}_{1.5y}\left(\frac{3}{16}aa_{0,0} + \frac{9}{16}aa_{0,1} + \frac{1}{16}a_{1,0} + \frac{3}{16}aa_{1,1}\right) \\ & + \mathrm{M}_{1.5x}\mathrm{M}_{1y}\left(\frac{3}{16}aa_{0,0} + \frac{1}{16}aa_{0,1} + \frac{9}{16}a_{1,0} + \frac{3}{16}aa_{1,1}\right) \\ & + \mathrm{M}_{1.5x}\mathrm{M}_{1.5y}\left(\frac{1}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{9}{16}aa_{1,1}\right) \\ & + \mathrm{M}_{1x}\mathrm{M}_{2y}\left(\frac{3}{4}ab_{0,2} + \frac{1}{4}ab_{1,2}\right) + \mathrm{M}_{1.5x}\mathrm{M}_{2y}\left(\frac{1}{4}ab_{0,2} + \frac{3}{4}ab_{1,2}\right) \\ & + \mathrm{M}_{2x}\mathrm{M}_{1y}\left(\frac{3}{4}ba_{2,0} + \frac{1}{4}ba_{2,1}\right) + \mathrm{M}_{2x}\mathrm{M}_{1.5y}\left(\frac{1}{4}ba_{2,0} + \frac{3}{4}ba_{2,1}\right) \\ & + bb_{2,2}\mathrm{M}_{2x}\mathrm{M}_{2y} \end{aligned} \tag{A.2}$$

In this case, there are eight constraints as listed in (A.3).

$$
\begin{aligned}
bb_{1,1} &= \frac{9}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{1}{16}aa_{1,1} \\
bb_{1.5,1} &= \frac{3}{16}aa_{0,0} + \frac{1}{16}aa_{0,1} + \frac{9}{16}a_{1,0} + \frac{3}{16}aa_{1,1} \\
bb_{1,1.5} &= \frac{3}{16}aa_{0,0} + \frac{9}{16}aa_{0,1} + \frac{1}{16}a_{1,0} + \frac{3}{16}aa_{1,1} \\
bb_{1.5,1.5} &= \frac{1}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{9}{16}aa_{1,1} \\
bb_{1,2} &= \frac{3}{4}ab_{0,2} + \frac{1}{4}ab_{1,2} \\
bb_{1.5,2} &= \frac{1}{4}ab_{0,2} + \frac{3}{4}ab_{1,2} \\
bb_{2,1} &= \frac{3}{4}ba_{2,0} + \frac{1}{4}ba_{2,1} \\
bb_{2,1.5} &= \frac{1}{4}ba_{2,0} + \frac{3}{4}ba_{2,1}
\end{aligned}
\tag{A.3}
$$

For $[2.0, 2.5] \times [2.5, 3.0]$, the active B-spline basis functions are N_{0x}, N'_{1x} and M_{2x} in the x direction and N_{0y}, N'_{1y}, M_{2y} and $\mathrm{M}_{2.5y}$ in the y direction. The spline surface, w, can be expressed as

$$
\begin{aligned}
w &= aa_{0,0}\mathrm{N}_{0x}\mathrm{N}_{0y} + aa_{0,1}\mathrm{N}_{0x}\mathrm{N}'_{1y} + ab_{0,2}\mathrm{N}_{0x}\mathrm{M}_{2y} + ab_{0,2.5}\mathrm{N}_{0x}\mathrm{M}_{2.5y} \\
&+ aa_{1,0}\mathrm{N}'_{1x}\mathrm{N}_{0y} + aa_{1,1}\mathrm{N}'_{1x}\mathrm{N}'_{1y} + ab_{1,2}\mathrm{N}'_{1x}\mathrm{M}_{2y} + ab_{1,2.5}\mathrm{N}'_{1x}\mathrm{M}_{2.5y} \\
&+ ba_{2,0}\mathrm{M}_{2x}\mathrm{N}_{0y} + ba_{2,1}\mathrm{M}_{2x}\mathrm{N}'_{1y} + bb_{2,2}\mathrm{M}_{2x}\mathrm{M}_{2y} + bb_{2,2.5}\mathrm{M}_{2x}\mathrm{M}_{2.5y} \\
&= \mathrm{M}_{1x}\mathrm{M}_{1.5y}\left(\frac{3}{16}aa_{0,0} + \frac{9}{16}aa_{0,1} + \frac{1}{16}a_{1,0} + \frac{3}{16}aa_{1,1}\right) \\
&+ \mathrm{M}_{1.5x}\mathrm{M}_{1.5y}\left(\frac{1}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{9}{16}aa_{1,1}\right) \\
&+ \mathrm{M}_{1x}\mathrm{M}_{2y}\left(\frac{3}{4}ab_{0,2} + \frac{1}{4}ab_{1,2}\right) + \mathrm{M}_{1.5x}\mathrm{M}_{2y}\left(\frac{1}{4}ab_{0,2} + \frac{3}{4}ab_{1,2}\right) \\
&+ \mathrm{M}_{1x}\mathrm{M}_{2.5y}\left(\frac{3}{4}ab_{0,2.5} + \frac{1}{4}ab_{1,2.5}\right) + \mathrm{M}_{1.5x}\mathrm{M}_{2.5y}\left(\frac{1}{4}ab_{0,2.5} + \frac{3}{4}ab_{1,2.5}\right) \\
&+ \mathrm{M}_{2x}\mathrm{M}_{1.5y}\left(\frac{1}{4}ba_{2,0} + \frac{3}{4}ba_{2,1}\right) \\
&+ bb_{2,2}\mathrm{M}_{2x}\mathrm{M}_{2y} + bb_{2,2.5}\mathrm{M}_{2x}\mathrm{M}_{2.5y}
\end{aligned}
\tag{A.4}
$$

And the seven constraints correspond to

$$
\begin{aligned}
bb_{1,1.5} &= \frac{3}{16}aa_{0,0} + \frac{9}{16}aa_{0,1} + \frac{1}{16}a_{1,0} + \frac{3}{16}aa_{1,1} \\
bb_{1.5,1.5} &= \frac{1}{16}aa_{0,0} + \frac{3}{16}aa_{0,1} + \frac{3}{16}a_{1,0} + \frac{9}{16}aa_{1,1} \\
bb_{1,2} &= \frac{3}{4}ab_{0,2} + \frac{1}{4}ab_{1,2} \\
bb_{1.5,2} &= \frac{1}{4}ab_{0,2} + \frac{3}{4}ab_{1,2} \\
bb_{1,2.5} &= \frac{3}{4}ab_{0,2.5} + \frac{1}{4}ab_{1,2.5} \\
bb_{1.5,2.5} &= \frac{1}{4}ab_{0,2.5} + \frac{3}{4}ab_{1,2.5} \\
bb_{2,1.5} &= \frac{1}{4}ba_{2,0} + \frac{3}{4}ba_{2,1}
\end{aligned}
\tag{A.5}
$$

As with adaptive mesh refinement associated with Lagrange basis functions, the refined region creates both dependent and independent nodes. Dependent nodes are formed when they are the hanging nodes across edges and/or faces and independent nodes are formed everywhere else. From (A.3) and (A.5), the coefficient $aa_{i,j}$ is associated with the basis functions on the coarse grid, which means that they are independent nodes. For $ab_{i,j}$ and $ba_{i,j}$, they depend on basis functions from both the refined and coarse grid, making them dependent nodes and they can be further expressed in terms of the independent nodes $aa_{i,j}$ from the coarse grid. For uniform element size in the refined domain, the dependent nodes $ab_{i,j}$ and $ba_{i,j}$ from (A.3)

and (A.5) can be simplified in terms of the independent nodes $aa_{i,j}$ as

$$\begin{aligned}
ab_{0,2} &= \frac{3}{4}aa_{0,1} + \frac{1}{4}aa_{0,2} \\
ab_{0,2.5} &= \frac{1}{4}aa_{0,1} + \frac{3}{4}aa_{0,2} \\
ab_{1,2} &= \frac{3}{4}aa_{1,1} + \frac{1}{4}aa_{1,2} \\
ab_{1,2.5} &= \frac{1}{4}aa_{1,1} + \frac{3}{4}aa_{1,2} \\
ba_{2,0} &= \frac{3}{4}aa_{1,0} + \frac{1}{4}aa_{2,0} \\
ba_{2,1} &= \frac{3}{4}aa_{1,1} + \frac{1}{4}aa_{2,1}
\end{aligned} \tag{A.6}$$

Substituting these relationships into (A.3) and (A.5),

$$\begin{aligned}
bb_{1,2} &= \frac{3}{4}ab_{0,2} + \frac{1}{4}ab_{1,2} \\
&= \frac{9}{16}aa_{0,1} + \frac{3}{16}aa_{0,2} + \frac{3}{16}a_{1,1} + \frac{1}{16}aa_{1,2} \\
bb_{1.5,2} &= \frac{1}{4}ab_{0,2} + \frac{3}{4}ab_{1,2} \\
&= \frac{3}{16}aa_{0,1} + \frac{1}{16}aa_{0,2} + \frac{9}{16}a_{1,1} + \frac{3}{16}aa_{1,2} \\
bb_{1,2.5} &= \frac{3}{4}ab_{0,2.5} + \frac{1}{4}ab_{1,2.5} \\
&= \frac{3}{16}aa_{0,1} + \frac{9}{16}aa_{0,2} + \frac{1}{16}a_{1,1} + \frac{3}{16}aa_{1,2} \\
bb_{1.5,2.5} &= \frac{1}{4}ab_{0,2.5} + \frac{3}{4}ab_{1,2.5} \\
&= \frac{1}{16}aa_{0,1} + \frac{3}{16}aa_{0,2} + \frac{3}{16}a_{1,1} + \frac{9}{16}aa_{1,2} \\
bb_{2,1} &= \frac{3}{4}ba_{2,0} + \frac{1}{4}ba_{2,1} \\
&= \frac{9}{16}aa_{1,0} + \frac{3}{16}aa_{2,0} + \frac{3}{16}a_{1,1} + \frac{1}{16}aa_{2,1} \\
bb_{2,1.5} &= \frac{1}{4}ba_{2,0} + \frac{3}{4}ba_{2,1} \\
&= \frac{3}{16}aa_{1,0} + \frac{1}{16}aa_{2,0} + \frac{9}{16}a_{1,1} + \frac{3}{16}aa_{2,1}
\end{aligned} \tag{A.7}$$

From (A.3) and (A.7), similarities can be found in the implementation of the

group of constraints $bb_{1,1}$, $bb_{1.5,1}$, $bb_{1,1.5}$ and $bb_{1.5,1.5}$ and $bb_{1,2}$, $bb_{1.5,2}$,$bb_{1,2.5}$ and $bb_{1.5,2.5}$. Refining the basis function N_{2x} and N_{3y}, the constraints on $bb_{1,2}$, $bb_{1.5,2}$,$bb_{1,2.5}$ and $bb_{1.5,2.5}$ can be explicitly formed in terms of $aa_{i,j}$ and are identical to (A.7).

Derivation of Subdivision Matrix for Bernstein Polynomials

The derivation of the subdivision matrix is required when using the Bernstein polynomials for adaptive mesh refinement or geometric multigrid as shown in (4.14). For the case of adaptive mesh refinement, we are interpolating across meshes with identical mesh order. For $p = q = 3$, the following expression holds

$$\begin{bmatrix} \mathrm{N}_0^c \\ \mathrm{N}_1^c \\ \mathrm{N}_2^c \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}_0^f \\ \mathrm{N}_1^f \\ \mathrm{N}_2^f \\ \mathrm{N}_3^f \\ \mathrm{N}_4^f \end{bmatrix} \tag{A.8}$$

For $p = q = 4$, the following expression holds

$$\begin{bmatrix} \mathrm{N}_0^c \\ \mathrm{N}_1^c \\ \mathrm{N}_2^c \\ \mathrm{N}_3^c \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{8} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3}{8} & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}_0^f \\ \mathrm{N}_1^f \\ \mathrm{N}_2^f \\ \mathrm{N}_3^f \\ \mathrm{N}_4^f \\ \mathrm{N}_5^f \\ \mathrm{N}_6^f \end{bmatrix} \tag{A.9}$$

For geometric multigrid, the subdivision matrix is used for interpolating across

meshes with different order. For $p = 3$ and $q = 4$,

$$\begin{bmatrix} \mathrm{N}_0^c \\ \mathrm{N}_1^c \\ \mathrm{N}_2^c \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & \frac{2}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}_0^f \\ \mathrm{N}_1^f \\ \mathrm{N}_2^f \\ \mathrm{N}_3^f \end{bmatrix} \tag{A.10}$$

For $p = 5$ and $q = 6$,

$$\begin{bmatrix} \mathrm{N}_0^c \\ \mathrm{N}_1^c \\ \mathrm{N}_2^c \\ \mathrm{N}_3^c \\ \mathrm{N}_4^c \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{2}{5} & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{5} & \frac{3}{5} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{N}_0^f \\ \mathrm{N}_1^f \\ \mathrm{N}_2^f \\ \mathrm{N}_3^f \\ \mathrm{N}_4^f \\ \mathrm{N}_5^f \end{bmatrix} \tag{A.11}$$

Bordering Method for Solving KKT System

The KKT system $\mathbf{J}_0\mathbf{p}_k = \mathbf{b}$ can be solved efficiently in parallel using a series of variable eliminations. This series of computation can be reduced to parallel vector operations and a small number of operations on small dense matrices. As a result, the factorization and the application of the factorization scales efficiently. The solution procedure begins by obtaining the solution for the slack variables and lower and upper Lagrange multiplier bound variables

$$\begin{aligned} \mathbf{p}_{z_l} &= \mathbf{P}^{-1}(\mathbf{b}_{z_l} - \mathbf{Z}_l\mathbf{p}_p), \\ \mathbf{p}_s &= \mathbf{Z}^{-1}(\mathbf{b}_s - \mathbf{S}\mathbf{p}_z), \\ \mathbf{p}_t &= \mathbf{Z}_t^{-1}(\mathbf{b}_{z_t} + \mathbf{T}(\mathbf{p}_z + \mathbf{b}_t)) \end{aligned} \tag{A.12}$$

Next, using the first two equations gives

$$\begin{aligned} b_0\mathbf{p}_p - \mathbf{A}^T\mathbf{p}_z - \mathbf{p}_{z_l} &= \mathbf{b}_p, \\ \mathbf{A}\mathbf{p}_p - \mathbf{p}_s + \mathbf{p}_t &= \mathbf{b}_z, \end{aligned} \tag{A.13}$$

Substituting the expressions for the slack and Lagrange multiplier updates (A.12) into the expression for the first two linearized KKT conditions (A.13) yields the following

$$\begin{aligned} \mathbf{D}\mathbf{p}_p - \mathbf{A}^T\mathbf{p}_z &= \mathbf{b}_p + \mathbf{P}^{-1}\mathbf{b}_{z_l}, \\ \mathbf{A}\mathbf{p}_p + (\mathbf{Z}^{-1}\mathbf{S} + \mathbf{Z}_t^{-1}\mathbf{T})\mathbf{p}_z &= \mathbf{b}_z + \mathbf{Z}^{-1}\mathbf{b}_s - \mathbf{Z}_t^{-1}(\mathbf{b}_{z_t} + \mathbf{T}\mathbf{b}_t) \end{aligned} \tag{A.14}$$

where the diagonal matrix $\mathbf{D}$ is defined as follows:

$$\mathbf{D} = \left[b_0\mathbf{I} + \mathbf{P}^{-1}\mathbf{Z}_l\right].$$

Finally, $\mathbf{p}_p$ can be eliminated for $\mathbf{p}_z$ as follows

$$(\mathbf{Z}^{-1}\mathbf{S} + \mathbf{Z}_t^{-1}\mathbf{T} + \mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T)\mathbf{p}_z = \mathbf{d}_z, \tag{A.15}$$

The right hand side $\mathbf{d}_z$ is

$$\mathbf{d}_z \triangleq \mathbf{b}_z + \mathbf{Z}^{-1}\mathbf{b}_s - \mathbf{Z}_t^{-1}(\mathbf{b}_{z_t} + \mathbf{T}\mathbf{b}_t) - \mathbf{A}\mathbf{D}^{-1}\left(\mathbf{b}_p + \mathbf{P}^{-1}\mathbf{b}_{z_l}\right).$$

This solution procedure is invoked each time a solution of the form $\mathbf{J}_0\mathbf{p}_k = \mathbf{b}$ is required.

REFERENCES

[1] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197 –224, 1988.

[2] L Krog, A Tucker, M. Kemp, and R. Boyd, "Toplogy Optimization of Wing Box Ribs," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, Sep. 1, 2004.

[3] S Grihon, L Krog, and K Hertel, "A380 weight savings using numerical structural optimization," in *Inproceedings of 20th AAAF Colloqium "Material for Aerospace Application"*, Paris, France, 2004.

[4] S. Buchanan, "Development of a wingbox rib for a passenger jet aircraft using design optimization and constrained to traditional design and manufacture requirements," in *CAE Technology Conference/Altair Engineering*, 2007.

[5] S. I. Kim, S. W. Kang, Y.-S. Yi, J. Park, and Y. Y. Kim, "Topology optimization of vehicle rear suspension mechanisms," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 8, pp. 1412–1433, 2018, NME-Aug-16-0622.R2.

[6] J. f. Shi and J. h. Sun, "Overview on innovation of topology optimization in vehicle cae," in *2009 International Conference on Electronic Computer Technology*, 2009, pp. 457–460.

[7] O. Sigmund and S. Torquato, "Design of materials with extreme thermal expansion using a three-phase topology optimization method," *Journal of the Mechanics and Physics of Solids*, vol. 45, no. 6, pp. 1037 –1067, 1997.

[8] A. R. Diaz and O. Sigmund, "A topology optimization method for design of negative permeability metamaterials," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 163–177, 2010.

[9] L. Lu, T. Yamamoto, M. Otomori, T. Yamada, K. Izui, and S. Nishiwaki, "Topology optimization of an acoustic metamaterial with negative bulk modulus using local resonance," *Finite Elements in Analysis and Design*, vol. 72, pp. 1 –12, 2013.

[10] O. Sigmund, "On the design of compliant mechanisms using topology optimization," *Mechanics of Structures and Machines*, vol. 25, no. 4, pp. 493–524, 1997.

[11] T. Bruns and D. Tortorelli, "Topology optimization of non-linear elastic structures and compliant mechanisms," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 2627, pp. 3443–3459, 2001.

[12] T. Borrvall and J. Petersson, "Topology optimization of fluids in stokes flow," *International Journal for Numerical Methods in Fluids*, vol. 41, no. 1, pp. 77–107, eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.426`.

[13] V. J. Challis and J. K. Guest, "Level set topology optimization of fluids in stokes flow," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 10, pp. 1284–1308, eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2616`.

[14] T. Zegard and G. H. Paulino, "Bridging topology optimization and additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 53, no. 1, pp. 175–192, 2016.

[15] D Brackett, I Ashcroft, and R Hague, "Topology optimization for additive manufacturing," in *Proceedings of the solid freeform fabrication symposium, Austin, TX*, S, vol. 1, 2011, pp. 348–362.

[16] N. Gardan and A. Schneider, "Topological optimization of internal patterns and support in additive manufacturing," *Journal of Manufacturing Systems*, vol. 37, no. Part 1, pp. 417 –425, 2015.

[17] I. Gibson, D. W. Rosen, and B. Stucker, "The use of multiple materials in additive manufacturing," in *Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing*. Boston, MA: Springer US, 2010, pp. 436–449, ISBN: 978-1-4419-1120-9.

[18] L. V. Gibiansky and O. Sigmund, "Multiphase composites with extremal bulk modulus," *Journal of the Mechanics and Physics of Solids*, vol. 48, no. 3, pp. 461 –498, 2000.

[19] J. Stegmann and E. Lund, "Discrete material optimization of general composite shell structures," *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 2009–2027, 14 2005.

[20] E. D. Sanders, M. A. Aguil, and G. H. Paulino, "Multi-material continuum topology optimization with arbitrary volume and mass constraints," *Computer Methods in Applied Mechanics and Engineering*, vol. 340, pp. 798 –823, 2018.

[21] C. Hvejsel and E. Lund, "Material interpolation schemes for unified topology and multi-material optimization," *Structural and Multidisciplinary Optimization*, vol. 43, no. 6, pp. 811–825, 2011.

[22] C. Hvejsel, E. Lund, and M. Stolpe, "Optimization strategies for discrete multi-material stiffness optimization," *Structural and Multidisciplinary Optimization*, vol. 44, no. 2, pp. 149–163, 2011.

[23] E. Lund, "Discrete material and thickness optimization of laminated composite structures including failure criteria," *Structural and Multidisciplinary Optimization*, 2017.

[24] G. J. Kennedy and J. R. R. A. Martins, "A laminate parametrization technique for discrete ply-angle problems with manufacturing constraints," *Structural and Multidisciplinary Optimization*, pp. 1–15, Jan. 1, 2013.

[25] T. Gao, P. Xu, and W. Zhang, "Topology optimization of thermo-elastic structures with multiple materials under mass constraint," *Computers & Structures*, vol. 173, pp. 150 –160, 2016.

[26] T. Gao and W. Zhang, "A mass constraint formulation for structural topology optimization with multiphase materials," *International Journal for Numerical Methods in Engineering*, vol. 88, no. 8, pp. 774–796, 2011.

[27] G. J. Kennedy, "Large-scale multimaterial topology optimization for additive manufacturing," in *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, FL, Jan. 1, 2015.

[28] M. P. Bendsøe and O. Sigmund, "Material interpolation schemes in topology optimization," *Archive of Applied Mechanics*, vol. 69, no. 9, pp. 635–654, 1999.

[29] M. Y. Wang and X. Wang, "colori level sets: A multi-phase method for structural topology optimization with multiple materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 6, pp. 469 –496, 2004.

[30] G. I. N. Rozvany, "A critical review of established methods of structural topology optimization," *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, pp. 217–237, 2009.

[31] E. A. Thornton, *Thermal structures for aerospace applications*. AIAA, 1996.

[32] A. K. Noor, S. L. Venneri, D. B. Paul, and M. A. Hopkins, "Structures technology for future aerospace systems," *Computers & Structures*, vol. 74, no. 5, pp. 507 –519, 2000.

[33] K. Sairajan, G. Aglietti, and K. Mani, "A review of multifunctional structure technology for aerospace applications," *Acta Astronautica*, vol. 120, no. Supplement C, pp. 30 –42, 2016.

[34] Z. Rao and S. Wang, "A review of power battery thermal energy management," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 9, pp. 4554 –4571, 2011.

[35] R. Mahamud and C. Park, "Reciprocating air flow for li-ion battery thermal management to improve temperature uniformity," *Journal of Power Sources*, vol. 196, no. 13, pp. 5685 –5696, 2011.

[36] T. Wang, K. Tseng, J. Zhao, and Z. Wei, "Thermal investigation of lithium-ion battery module with different cell arrangement structures and forced air-cooling strategies," *Applied Energy*, vol. 134, no. Supplement C, pp. 229 –238, 2014.

[37] R. Grandhi, "Structural optimization with frequency constraints-a review," *AIAA journal*, vol. 31, no. 12, pp. 2296–2303, 1993.

[38] P. Solin, J. Cerveny, L. Dubcova, and D. Andrs, "Monolithic discretization of linear thermoelasticity problems via adaptive multimesh hp-fem," *Journal of Computational and Applied Mathematics*, vol. 234, no. 7, pp. 2350 –2357, 2010, Fourth International Conference on Advanced Computational Methods in ENgineering (ACOMEN 2008).

[39] F. Larsson, P. Hansbo, and K. Runesson, "Space-time finite elements and an adaptive strategy for the coupled thermoelasticity problem," *International Journal for Numerical Methods in Engineering*, vol. 56, no. 2, pp. 261–293, 2003.

[40] D. Aubry, D. Lucas, and B. Tie, "Adaptive strategy for transient/coupled problems applications to thermoelasticity and elastodynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 176, no. 1, pp. 41 –50, 1999.

[41] G. L. Sleijpen and H. A. Van der Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM review*, vol. 42, no. 2, pp. 267–293, 2000.

[42] M. Hochstenbach and Y. Notay, "The JacobiDavidson method," *GAMM-Mitteilungen*, vol. 29, no. 2, pp. 368–382, 2006.

[43] H. Voss, "A new justification of the jacobidavidson method for large eigenproblems," *Linear Algebra and its Applications*, vol. 424, no. 2, pp. 448 –455, 2007.

[44] R. G. Grimes, J. G. Lewis, and H. D. Simon, "A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 1, pp. 228–272, 1994.

[45] N. Aage and B. S. Lazarov, "Parallel framework for topology optimization using the method of moving asymptotes," *Structural and Multidisciplinary Optimization*, vol. 47, no. 4, pp. 493–505, 2013.

[46] N. Aage, E. Andreassen, and B. Lazarov, "Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework," *Structural and Multidisciplinary Optimization*, vol. 51, no. 3, pp. 565–572, 2015.

[47] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, "Giga-voxel computational morphogenesis for structural design," *Nature.*, vol. 550, no. 7674, pp. 84–86, 2017.

[48] Wang Shun and Sturler Eric de and Paulino Glaucio H., "Largescale topology optimization using preconditioned krylov subspace methods with recycling," *International Journal for Numerical Methods in Engineering*, vol. 69, no. 12, pp. 2441–2468,

[49] A. Evgrafov, C. J. Rupp, K. Maute, and M. L. Dunn, "Large-scale parallel topology optimization using a dual-primal substructuring solver," *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 329–345, 2008.

[50] T. Borrvall and J. Petersson, "Large-scale topology optimization in 3d using parallel computing," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 4647, pp. 6201 –6229, 2001.

[51] J. Bennett and M. Botkin, "Structural shape optimization with geometric description and adaptive mesh refinement," *AIAA journal*, vol. 23, no. 3, pp. 458–464, 1985.

[52] K. Noboru, K. Y. Chung, T. Toshikazu, and J. E. Taylor, "Adaptive finite element methods for shape optimization of linearly elastic structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 57, no. 1, pp. 67 –89, 1986.

[53] M. Botkin, "Shape optimization with multiple loading conditions and mesh refinement," *AIAA journal*, vol. 28, no. 5, pp. 922–927, 1990.

[54] J. C. A. Costa Jr and M. K. Alves, "Layout optimization with h-adaptivity of structures," *International Journal for Numerical Methods in Engineering*, vol. 58, no. 1, pp. 83–102, 2003.

[55] R. Stainko, "An adaptive multilevel approach to the minimal compliance problem in topology optimization," *Communications in Numerical Methods in Engineering*, vol. 22, no. 2, pp. 109–118, 2006.

[56] K. Maute and E. Ramm, "Adaptive topology optimization," *Structural optimization*, vol. 10, no. 2, pp. 100–112, 1995.

[57] K. Maute and E. Ramm, "Adaptive topology optimization of shell structures," *AIAA journal*, vol. 35, no. 11, pp. 1767–1773, 1997.

[58] J. K. Guest and L. C. Smith Genut, "Reducing dimensionality in topology optimization using adaptive design variable fields," *International Journal for Numerical Methods in Engineering*, vol. 81, no. 8, pp. 1019–1045, 2010.

[59] Y. Wang, Z. Kang, and Q. He, "An adaptive refinement approach for topology optimization based on separated density field description," *Computers & Structures*, vol. 117, pp. 10 –22, 2013.

[60] A. B. Lambe and A. Czekanski, "Topology optimization using a continuous density field and adaptive mesh refinement," *International Journal for Numerical Methods in Engineering*, n/a–n/a, nme.5617.

[61] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineerng analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 337–357, 1987.

[62] M. Bruggi and M. Verani, "A fully adaptive topology optimization algorithm with goal-oriented error control," *Computers & Structures*, vol. 89, no. 1516, pp. 1481 –1493, 2011.

[63] E. de Sturler, G. H. Paulino, and S. Wang, "Topology optimization with adaptive mesh refinement," in *6th International Conference on Computation of Shell and Spatial Structures*, Ithaca, New York, May 28, 2008.

[64] Y. Wang, Z. Kang, and Q. He, "Adaptive topology optimization with independent error control for separated displacement and density fields," *Computers & Structures*, vol. 135, pp. 50 –61, 2014.

[65] H. Rodrigues and P. Fernandes, "A material based model for topology optimization of thermoelastic structures," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 12, pp. 1951–1965, 1995.

[66] Q. Li, G. P. Steven, and Y. Xie, "Displacement minimization of thermoelastic structures by evolutionary thickness design," *Computer Methods in Applied Mechanics and Engineering*, vol. 179, no. 34, pp. 361 –378, 1999.

[67] S. Cho and J.-Y. Choi, "Efficient topology optimization of thermo-elasticity problems using coupled field adjoint sensitivity analysis method," *Finite Elements in Analysis and Design*, vol. 41, no. 15, pp. 1481 –1495, 2005.

[68] Q. Xia and M. Y. Wang, "Topology optimization of thermoelastic structures using level set method," *Computational Mechanics*, vol. 42, no. 6, p. 837, 2008.

[69] J. D. Deaton and R. V. Grandhi, "Stiffening of restrained thermal structures via topology optimization," *Structural and Multidisciplinary Optimization*, vol. 48, no. 4, pp. 731–745, 2013.

[70] S. Deng and K. Suresh, "Topology optimization under thermo-elastic buckling," *Structural and Multidisciplinary Optimization*, vol. 55, no. 5, pp. 1759–1772, 2017.

[71] B. Stanford and P. Beran, "Aerothermoelastic topology optimization with flutter and buckling metrics," *Structural and Multidisciplinary Optimization*, vol. 48, no. 1, pp. 149–171, 2013.

[72] C. Jog, "Distributed-parameter optimization and topology design for nonlinear thermoelasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 132, no. 1, pp. 117 –134, 1996.

[73] P. Pedersen and N. L. Pedersen, "Strength optimized designs of thermoelastic structures," *Structural and Multidisciplinary Optimization*, vol. 42, no. 5, pp. 681–691, 2010.

[74] ——, "Interpolation/penalization applied for strength design of 3d thermoelastic structures," *Structural and Multidisciplinary Optimization*, vol. 45, no. 6, pp. 773–786, 2012.

[75] J. D. Deaton and R. V. Grandhi, "Stress-based design of thermal structures via topology optimization," *Structural and Multidisciplinary Optimization*, vol. 53, no. 2, pp. 253–270, 2016.

[76] S. N. Sørensen, R. Sørensen, and E. Lund, "DMTO a method for discrete material and thickness optimization of laminated composite structures," *Structural and Multidisciplinary Optimization*, vol. 50, no. 1, pp. 25–47, 2014.

[77] S. N. Sørensen and E. Lund, "Topology and thickness optimization of laminated composites including manufacturing constraints," *Structural and Multidisciplinary Optimization*, vol. 48, no. 2, pp. 249–265, 2013.

[78] M. Bruyneel, P. Duysinx, C. Fleury, and T. Gao, "Extensions of the shape functions with penalization parameterization for composite-ply optimization," *AIAA Journal*, vol. 49, no. 10, pp. 2325–2329, 2011.

[79] M. Bruyneel, "SFP : A new parameterization based on shape functions for optimal material selection: Application to conventional composite plies," *Structural and Multidisciplinary Optimization*, vol. 43, no. 1, pp. 17–27, 2011.

[80] Y. Wang, Z. Luo, Z. Kang, and N. Zhang, "A multi-material level set-based topology and shape optimization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 283, no. Supplement C, pp. 1570 –1586, 2015.

[81] K. A. James, "Multiphase topology design with optimal material selection using an inverse p-norm function," *International Journal for Numerical Methods in Engineering*, vol. 114, no. 9, pp. 999–1017, 2018.

[82] G. J. Kennedy and T. W. Chin, "A sequential convex optimization method for multimaterial compliance design problems," *Computers & Structures*, vol. 212, pp. 110 –124, 2019.

[83] O. Sigmund, "Design of multiphysics actuators using topology optimization part ii: two-material structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 49, pp. 6605 –6627, 2001.

[84] T. W. Chin and G. J. Kennedy, "Multimaterial thermoelastic stress-constrained topology optimization of structures with adaptive mesh refinement," in *2018 Multidisciplinary Analysis and Optimization Conference*, Jun. 30, 2018, published.

[85] Z. Kang and K. A. James, "Multimaterial topology design for optimal elastic and thermal response with material-specific temperature constraints," *International Journal for Numerical Methods in Engineering*, vol. 0, no. ja,

[86] W. F. Mitchell, "How high a degree is high enough for high order finite elements?" *Procedia Computer Science*, vol. 51, pp. 246 –255, 2015, International Conference On Computational Science, ICCS 2015.

[87] A. Díaz and O. Sigmund, "Checkerboard patterns in layout optimization," *Structural optimization*, vol. 10, no. 1, pp. 40–45, 1995.

[88] C. S. Jog and R. B. Haber, "Stability of finite element models for distributed-parameter optimization and topology design," *Computer Methods in Applied Mechanics and Engineering*, vol. 130, no. 3, pp. 203 –226, 1996.

[89] B. S. Lazarov and O. Sigmund, "Filters in topology optimization based on helmholtz-type differential equations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 6, pp. 765–781,

[90] O. Sigmund and J. Petersson, "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Structural optimization*, vol. 16, no. 1, pp. 68–75, 1998.

[91] K. Matsui and K. Terada, "Continuous approximation of material distribution for topology optimization," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 14, pp. 1925–1944,

[92] Z. Kang and Y. Wang, "Structural topology optimization based on non-local shepard interpolation of density field," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49, pp. 3515 –3525, 2011.

[93] A. V. Kumar and A. Parthasarathy, "Topology optimization using b-spline finite elements," *Structural and Multidisciplinary Optimization*, vol. 44, no. 4, pp. 471–481, 2011.

[94] X. Qian, "Topology optimization in b-spline space," *Computer Methods in Applied Mechanics and Engineering*, vol. 265, pp. 15 –35, 2013.

[95] A. B. Lambe and A. Czekanski, "A density field parametrization for topology optimization using bernstein elements," *International Journal for Numerical Methods in Engineering*, vol. 115, no. 10, pp. 1266–1286,

[96] A. P. Seyranian, E. Lund, and N. Olhoff, "Multiple eigenvalues in structural optimization problems," *Structural and Multidisciplinary Optimization*, vol. 8, no. 4, pp. 207–227, 1994.

[97] N. Olhoff, "Multicriterion structural optimization via bound formulation and mathematical programming," *Structural and Multidisciplinary Optimization*, vol. 1, pp. 11–17, 1 1989.

[98] J. Du and N. Olhoff, "Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps," *Structural and Multidisciplinary Optimization*, vol. 34, no. 2, pp. 91–110, 2007.

[99] T. S. Kim and Y. Y. Kim, "Mac-based mode-tracking in structural topology optimization," *Computers & Structures*, vol. 74, no. 3, pp. 375–383, 2000.

[100] Y Maeda, S. Nishiwaki, K Izui, M Yoshimura, K Matsui, and K Terada, "Structural topology optimization of vibrating structures with specified eigenfrequencies and eigenmode shapes," *International Journal for Numerical Methods in Engineering*, vol. 67, no. 5, pp. 597–628, 2006.

[101] X. Chen, H. Qi, L. Qi, and K.-L. Teo, "Smooth convex approximation to the maximum eigenvalue function," *Journal of Global Optimization*, vol. 30, no. 2, pp. 253–270, 2004.

[102] Z.-D. Ma, N. Kikuchi, and H.-C. Cheng, "Topological design for vibrating structures," *Computer methods in applied mechanics and engineering*, vol. 121, no. 1-4, pp. 259–280, 1995.

[103] T. W. Chin and G. J. Kennedy, "Large-scale compliance-minimization and buckling topology optimization of the undeformed common research model wing," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech*, Jan. 1, 2016, published.

[104] P. D. Dunning, E. Ovtchinnikov, J. Scott, and H. A. Kim, "Level-set topology optimization with many linear buckling constraints using an efficient and robust eigensolver," *International Journal for Numerical Methods in Engineering*, vol. 107, no. 12, pp. 1029–1053, 2016.

[105] E. Andreassen, F. Ferrari, O. Sigmund, and A. R. Diaz, "Frequency response as a surrogate eigenvalue problem in topology optimization," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 8, pp. 1214–1229, 2017.

[106] F. Federico, L. B. S., and S. Ole, "Eigenvalue topology optimization via efficient multilevel solution of the frequency response," *International Journal for Numerical Methods in Engineering*, vol. 115, no. 7, pp. 872–892, 2018.

[107] G. J. Kennedy and J. R. R. A. Martins, "A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures," *Finite Elements in Analysis and Design*, vol. 87, no. 0, pp. 56 –73, Jan. 1, 2014.

[108] O. Amir, N. Aage, and B. Lazarov, "On multigrid-CG for efficient topology optimization," *Structural and Multidisciplinary Optimization*, vol. 49, no. 5, pp. 25–57, 2014.

[109] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[110] K. Svanberg, "The method of moving asymptotes: a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, 1987.

[111] S. Rojas-Labanda and M. Stolpe, "Benchmarking optimization solvers for structural topology optimization," *Structural and Multidisciplinary Optimization*, vol. 52, no. 3, pp. 527–547, 2015.

[112] R. Fletcher, "A model algorithm for composite nondifferentiable optimization problems," in *Nondifferential and Variational Techniques in Optimization*, D. C. Sorensen and R. J.-B. Wets, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 67–76.

[113] R. H. Byrd, J. Nocedal, and R. B. Schnabel, "Representations of quasi-Newton matrices and their use in limited memory methods," *Mathematical Programming*, vol. 63, no. 1-3, pp. 129–156, 1994.

[114] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust region methods*. SIAM, 2000, vol. 1.

[115] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd, ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[116] C. Burstedde, L. C. Wilcox, and O. Ghattas, "P4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1103–1133, 2011. eprint: `http://dx.doi.org/10.1137/100791634`.

[117] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[118] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129 –147, 1982.

[119] Open CASCADE S. A. S. (2010). Open CASCADE Technology, 3D modeling & numerical simulation, (visited on 01/30/2017).

[120] S. J. Owen, "A survey of unstructured mesh generation technology.," in *IMR*, 1998, pp. 239–267.

[121] J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzainet, "Blossom-quad: a non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 9, pp. 1102–1119, 2012.

[122] C. Geuzaine and J.-F. Remacle, "Gmsh: a 3-d finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[123] J Edmonds, E. Johnson, and S. Lockhart, "Blossom I: a computer code for the matching problem," *IBM TJ Watson Research Center, Yorktown Heights, New York*, 1969.

[124] J. Edmonds, "Maximum matching and a polyhedron with $0, 1$ vertices," *J. of Res. the Nat. Bureau of Standards*, vol. 69, pp. 125–130,

[125] V. Kolmogorov, "Blossom v: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, 2009.

[126] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 359–392, 1998.

[127] P. M. Knupp, "Algebraic mesh quality metrics for unstructured initial meshes," *Finite Elements in Analysis and Design*, vol. 39, no. 3, pp. 217 –241, 2003.

[128] Sandia National Laboratories. (2018). Cubit, the sandia national laboratory automated mesh generation toolkit, (visited on 07/2018).

[129] M. Stople and K. Svanberg, "An alternative interpolation scheme for minimum compliance topology optimization," *Structural and Multidisciplinary Optimization*, vol. 22, pp. 116–124, 2001.

[130] J. K. Guest, J. H. Prvost, and T. Belytschko, "Achieving minimum length scale in topology optimization using nodal design variables and projection functions," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 2, pp. 238–254,

[131] M. R. Eslami, R. B. Hetnarski, J. Ignaczak, N. Noda, N. Sumi, and Y. Tanigawa, "Finite element of coupled thermoelasticity," in *Theory of Elasticity and Thermal Stresses: Explanations, Problems and Solutions*. Dordrecht: Springer Netherlands, 2013, pp. 727–753.

[132] T. Gao and W. Zhang, "Topology optimization involving thermo-elastic stress loads," *Structural and Multidisciplinary Optimization*, vol. 42, no. 5, pp. 725–738, 2010.

[133] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli, "Stress-based topology optimization for continua," *Structural and Multidisciplinary Optimization*, vol. 41, no. 4, pp. 605–620, 2010.

[134] G. J. Kennedy and J. E. Hicken, "Improved constraint-aggregation methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 289, no. 0, pp. 332 –354, 2015.

[135] G. J. Kennedy, "Strategies for adaptive optimization with aggregation constraints using interior-point methods," *Computers & Structures*, vol. 153, no. 0, pp. 217 –229, 2015.

[136] S. P. Software. (2017). Siemens nx 12.0, (visited on 09/20/2018).

[137] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.

[138] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*. Springer-Verlag New York, 1978, vol. 27.

[139] K. Höllig, *Finite element methods with B-splines*. Siam, 2003, vol. 26.

[140] K. A. Johannessen, F. Remonato, and T. Kvamsdal, "On the similarities and differences between classical hierarchical, truncated hierarchical and {lr} b-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 64 –101, 2015.

[141] W. Jiang and J. E. Dolbow, "Adaptive refinement of hierarchical b-spline finite elements with an efficient data transfer algorithm," *International Journal for Numerical Methods in Engineering*, vol. 102, no. 3-4, pp. 233–256, 2015.

[142] P. Bornemann and F Cirak, "A subdivision-based implementation of the hierarchical b-spline finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 584–598, 2013.

[143] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes, "Polytop: a matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes," *Structural and Multidisciplinary Optimization*, vol. 45, no. 3, pp. 329–357, 2012.

[144] R. T. Farouki, "The bernstein polynomial basis: a centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379 –419, 2012.

[145] A. J. Torii and J. R. d. Faria, "Structural optimization considering smallest magnitude eigenvalues: A smooth approximation," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 39, no. 5, pp. 1745–1754, 2017.

[146] P. Salas, L. Giraud, Y. Saad, and S. Moreau, "Spectral recycling strategies for the solution of nonlinear eigenproblems in thermoacoustics," *Numerical Linear Algebra with Applications*, vol. 22, no. 6, pp. 1039–1058, 2015.

[147] M. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti, "Recycling krylov subspaces for sequences of linear systems," *SIAM Journal on Scientific Computing*, vol. 28, no. 5, pp. 1651–1674, 2006.

[148] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd. SIAM, 2003.

[149] K. Stben, "A review of algebraic multigrid," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1, pp. 281 –309, 2001, Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[150] T. Buhl, C. Pedersen, and O. Sigmund, "Stiffness design of geometrically nonlinear structures using topology optimization," *Structural and Multidisciplinary Optimization*, vol. 19, no. 2, pp. 93–104, 2000.

[151] H. Chung, O. Amir, and H. A. Kim, 0, 2019.